

A single-phase, proximal path-following framework*

Quoc Tran-Dinh

Department of Statistics and Operations Research, UNC-Chapel Hill, USA, quocdt@email.unc.edu

Anastasios Kyrillidis

University of Texas at Austin, USA, anastasios@utexas.edu

Volkan Cevher

Laboratory for Information and Inference Systems (LIONS), EPFL, Switzerland, volkan.cevher@epfl.ch

We propose a new proximal, path-following framework for a class of constrained convex problems. We consider settings where the nonlinear—and possibly non-smooth—objective part is endowed with a proximity operator, and the constraint set is equipped with a self-concordant barrier. Our approach relies on the following two main ideas. First, we re-parameterize the optimality condition as an auxiliary problem, such that a good initial point is available; by doing so, a family of alternative paths towards the optimum is generated. Second, we combine the proximal operator with path-following ideas to design a single-phase, proximal, path-following algorithm. Our method has several advantages. First, it allows handling non-smooth objectives via proximal operators; this avoids lifting the problem dimension in order to accommodate non-smooth components in optimization. Second, it consists of only a *single phase*: While the overall convergence rate of classical path-following schemes for self-concordant objectives does not suffer from the initialization phase, proximal path-following schemes undergo slow convergence, in order to obtain a good starting point [50]. In this work, we show how to overcome this limitation in the proximal setting and prove that our scheme has the same $\mathcal{O}(\sqrt{\nu} \log(1/\varepsilon))$ worst-case iteration-complexity with standard approaches [33, 37] without requiring an initial phase, where ν is the barrier parameter and ε is a desired accuracy. Finally, our framework allows errors in the calculation of proximal-Newton directions, without sacrificing the worst-case iteration complexity. We demonstrate the merits of our algorithm via three numerical examples, where proximal operators play a key role.

Key words: Proximal-Newton method, path-following schemes, non-smooth convex optimization.

MSC2000 subject classification: 90C06; 90C25; 90-08

OR/MS subject classification: Interior-point methods, non-smooth convex programming

1. Introduction. This paper studies the following constrained convex optimization problem:

$$G^* := \min_{x \in \mathbb{R}^p} \left\{ G(x) := \langle c, x \rangle + g(x) : x \in \mathcal{X} \right\}, \quad (1)$$

where $c \in \mathbb{R}^p$, g is a possibly non-smooth, proper, closed and convex function from \mathbb{R}^p to $\mathbb{R} \cup \{+\infty\}$ and \mathcal{X} is a nonempty, closed and convex set in \mathbb{R}^p .¹ We denote by \mathcal{X}^* the optimal solution set of (1), and by x^* an optimal solution in \mathcal{X}^* .

For convex sets \mathcal{X} , associated with a self-concordant barrier (see Section 2 for details), and for G *self-concordant and smooth*, e.g., just linear or quadratic, interior point methods (IPMs) often constitute the method-of-choice for solving (1), with a well-characterized worst-case complexity. A non-exhaustive list of instances of (1) includes linear programs, quadratic programs, second order cone programs, semi-definite programs, and geometric optimization [2, 7, 8, 17, 31, 32, 33, 38, 42, 44, 53, 54].

At the heart of IPMs lies the notion of *interior barriers*: these mimic the effect of the constraint set \mathcal{X} in (1) by appropriately penalizing the objective function with a barrier f over the set \mathcal{X} , as follows:

$$F_t^* := \min_{x \in \text{int}(\mathcal{X})} \left\{ F_t(x) := \frac{1}{t} \cdot G(x) + f(x) \right\}. \quad (2)$$

* This is the first revision. The initial version was uploaded on arxiv on March 5, 2016.

¹ We note that the linear term $\langle c, x \rangle$ can be absorbed into g . However, we separate it from g for our convenience in processing numerical examples in the last section.

Here, f models the structure of the feasible set \mathcal{X} and $t > 0$ is a penalty parameter. For different values of t , the regularized problem generates a sequence of solutions $\{x^*(t) : t > 0\}$, known as the *central path*, converging to x^* of (1), as t goes to 0^+ . *Path-following methods* operate along the central path: for a properly decreasing sequence of t values, they solve (2) only approximately, by performing a few Newton iterations for each t value; standard path-following schemes even perform just *one* Newton iteration, assuming a linear objective $G(x) := \langle c, x \rangle$ with no non-smooth term $g(x)$. For such problem cases, this is sufficient to guarantee that the approximate solution lies sufficiently close to the central path, and operates as warm-start for the next value of t in (2) [8, 30, 37, 33]. One requirement is that the initial point must lie within a predefined neighborhood of the central path. In their seminal work [37], Nesterov and Nemirovski showed that such methods admit a polynomial worst-case complexity, as long as the Newton method has polynomial complexity.

Based on the above, standard schemes [31, 33, 37] can be characterized by two phases: PHASE I and PHASE II. In PHASE I and for an initial value of t , say t_0 , one has to solve (2) carefully in order to determine a good initial point for PHASE II; this implies solving (2) up to sufficient accuracy, such that the Newton method for (2) admits fast convergence. In PHASE II and using the output of PHASE I as a warm-start, we path-follow with a provably polynomial time complexity.

Taking into account both phases, standard path-following algorithms—where (2) is a *self-concordant* objective—are characterized by the following iteration complexity. The total number of iterations required to obtain an ε -solution is

$$\mathcal{O}\left(\sqrt{\nu} \log\left(\frac{1}{\varepsilon}\right)\right). \quad (3)$$

Here, ν is a barrier parameter (see Section 2 for details) and ε is the approximate parameter, according to the following definition:

DEFINITION 1. Given a tolerance $\varepsilon > 0$, we say that x_ε^* is an ε -solution for (1) if

$$x_\varepsilon^* \in \mathcal{X}, \quad \text{and} \quad G(x_\varepsilon^*) - G^* \leq \varepsilon.$$

1.1. Path-following schemes for non-smooth objectives. For many applications in machine learning, optimization and signal processing [8, 40, 50], the g part in (1) could be non-smooth (or even smooth but non-self-concordant). Such a g term is usually included in the optimization in order to leverage the true underlying structure in x^* . An example is the ℓ_1 -norm regularization, *i.e.*, $g(x) = \|x\|_1$, with applications in high-dimensional statistics, compressive sensing, scientific and medical imaging [12, 18, 20, 23, 29, 41, 47, 55], among others. Other examples for g include the indicator function of a convex set [40], the $\ell_{1,2}$ -group norm [4, 22, 24], and the nuclear norm [11] using in low-rank matrix approximation.

Unfortunately, non-smoothness in the objective reduces the optimization efficiency. In such settings, one can often reformulate (1) into a standard conic program, by introducing slack variables and additional constraints to model g . Such a technique is known as *disciplined convex programming* (DCP) [19] and has been incorporated in well-known software packages, such as CVX [19] and YALMIP [28]. Existing off-the-shelf solvers are then utilized to solve the resulting problem. However, DCP could potentially increase the problem dimension significantly; this, in sequence, reduces the efficiency of the IPMs. For instance, in the example above where $g(x) = \|x\|_1$, DCP introduces p slack variables to reformulate g into $\mathcal{O}(p)$ additional linear constraints; when $g(X) = \|X\|_*$, *i.e.*, g is the nuclear norm (sum of singular values of $X \in \mathbb{R}^{p \times q}$), then it can be *smoothed* via a semi-definite formulation, where the memory requirements and the volume of computation per iteration are high [27].

In this paper, we focus on cases where g is endowed with a generalized proximity operator, associated with a local norm $\|\cdot\|_x$ (see Section 2 for details):

$$\text{prox}_g(u) := \arg \min_{v \in \mathbb{R}^p} \left\{ g(v) + 1/2 \cdot \|v - u\|_x^2 \right\}.$$

Such proximity operators have been used extensively in non-smooth optimization problems, and proven to be efficient in real applications, under common gradient Lipschitz-continuity and strong convexity assumptions on the objective function [6, 13, 33]. However, for generic \mathcal{X} constraints in (1), the resulting interior barrier f in (2) does not have Lipschitz continuous gradients and, thus, prevents us from trivially recycling such ideas. This necessitates the design of a new class of path-following schemes, that exploit proximal operators and thus can accommodate non-smooth terms in the objective.

To the best of our knowledge, [50] is the first work that treats jointly interior barrier path-following schemes and proximity operators, in order to construct new proximal path-following algorithms for problems as in (1). According to [50], the proposed algorithm follows a two-phase approach, with PHASE II having the same worst-case iteration-complexity as in (3) (up to constants) [33, 37]. However, the initialization PHASE I in [50] requires substantial computational effort, which usually dominates the overall computational time. In particular, to find a good initial point, [50] uses a damped-step proximal-Newton scheme for (2), starting from an arbitrary initial point x^0 and for arbitrary selected $t_0 > 0$. For such configuration, [50] requires

$$\left\lceil \frac{F_{t_0}(x^0) - F_{t_0}(x^*(t_0))}{\omega((1 - \kappa)\beta)} \right\rceil$$

damped-step Newton iterations in PHASE I in order to find a point close to the optimal solution of (2), say $x^*(t_0)$, for the selected t_0 . Here, $\kappa \in (0, 1)$, $\beta \in (0, 0.15]$, and $\omega(\tau) := \tau - \log(1 + \tau) \geq 0$; see [50, Theorem 4.4] for more details. *I.e.*, in stark contrast to the global iteration complexity (3) of smooth path-following schemes, PHASE I of [50] might require a substantial number of iterations just to converge to a point close to the central path, and depends on the arbitrary initial point selection x^0 .

1.2. Motivation. From our discussion so far, it is clear that most existing works on path-following schemes require two phases. In the case of *smooth* self-concordant objectives in (1), PHASE I is often implemented as a damped-step Newton scheme, which has sublinear convergence rate, or an auxiliary path-following scheme, with linear convergence rate that satisfies the global, worst-case complexity in (3) [33, 37]. In standard conic programming, one can unify a two-phase algorithm in a single-phase IP path-following scheme via homogeneous and self-dual embedded strategies; see, *e.g.*, [45, 52, 54]. Such strategies parameterize the KKT condition of the primal and dual conic program so that one can immediately have an initial point, without performing PHASE I. So far and to the best of our knowledge, it remains unclear how such an auxiliary path-following scheme can find an initial point for *non-smooth* objectives in (1).

1.3. Our contributions. The goal of this paper is to develop a new single-phase, proximal path-following algorithm for (1). To do so, we first re-parameterize the optimality condition of the barrier problem associated with (1) as a *parametric monotone inclusion* (PMI). Then, we design a proximal path-following scheme to approximate the solution of such PMI, while controlling the penalty parameter. Finally, we show how to recover an approximate solution of (1), from the approximate solution of the PMI.

The main contributions of this paper can be summarized as follows:

(i) We introduce a new parameterization for the optimality condition of (2) to appropriately select the parameters such that less computation for initialization is needed. Thus, with an appropriate choice of parameters, we show how we can eliminate the slowly-convergent PHASE I in [50], while we still maintain the global, polynomial time, worst-case iteration-complexity.

In particular, we propose novel—checkable *a priori*—conditions over the set of initial points that can achieve the desiderata; this, in turn, provides rigorous configurations of the algorithm’s parameters such that the worst-case iteration complexity guarantee is obtained provably, avoiding the slowly convergent initialization procedures proposed so far for non-smooth optimization in (1).

(ii) We design a single-phase, path-following algorithm to compute an ε -solution of (1). For each t value, the resulting algorithm only requires a *single approximate Newton iteration* (see [50]), followed by a proximal step, of a strongly convex quadratic composite subproblem. We will use the term *proximal Newton step* when referring to these two steps. The algorithm allows inexact Newton steps, with a verifiable stopping criterion (cf. eq. (25)).

In particular, we establish the following result:

THEOREM 1. *The total number of proximal Newton iterations required, in order to reach an ε -solution of (1), is upper bounded by $\mathcal{O}\left(\sqrt{\nu} \log\left(\frac{\nu}{\varepsilon}\right)\right)$.*

A complete and formal description of the above theorem and its proof are provided in Section 4. Our *proximal* algorithm admits the same iteration-complexity, as standard path-following methods [33, 37] (up to a constant). To highlight the iteration complexity gains from the two-phase algorithm in [50, Theorem 4.4], recall that in the latter case, the total number of proximal Newton steps are bounded by:

$$\left\lceil \frac{F_{t_0}(x^0) - F_{t_0}(x_{t_0}^*)}{\omega((1-\kappa)\beta)} \right\rceil + \mathcal{O}\left(\sqrt{\nu} \log\left(\frac{\nu}{\varepsilon}\right)\right),$$

where the first term is in PHASE I as mentioned previously, and the second one is in PHASE II.

Our algorithm requires a well-chosen initial point that avoids PHASE I; one such case is that of an approximation of the analytical center \bar{x}_f^* of the barrier f (see Section 2 for details). In the text, we argue that evaluating this point is much easier than finding an initial point x^0 using PHASE I, as in [50]. In addition, for many feasible sets \mathcal{X} in (1), we can explicitly and easily compute \bar{x}_f^* of f (see Section 5 for examples).

1.4. The structure of the paper. This paper is organized as follows. Sections 2 and 3 contain basic definitions and notions, used in our analysis. We introduce a new re-parameterization of the central path in order to obtain a *predefined* initial point. Section 4 presents a novel algorithm and its complexity theory for the non-smooth objective function. Section 5 provides three numerical examples that highlight the merits of our algorithm.

2. Preliminaries. In this section, we provide the basic notation used in the rest of the paper, as well as two key concepts: proximity operators and self-concordant (barrier) functions.

2.1. Basic definitions. Given $x, y \in \mathbb{R}^p$, we use $\langle x, y \rangle$ or $x^T y$ to denote the inner product in \mathbb{R}^p . For a proper, closed and convex function g , we denote by $\text{dom}(g)$ its domain, (i.e., $\text{dom}(g) := \{x \in \mathbb{R}^p : g(x) < +\infty\}$), and by $\partial g(x) := \{v \in \mathbb{R}^p | g(y) \geq g(x) + \langle v, y - x \rangle, \forall y \in \text{dom}(g)\}$ its subdifferential at x . We also denote by $\text{Dom}(g) := \text{cl}(\text{dom}(g))$ the closure of $\text{dom}(g)$ [43]. We use $\mathcal{C}^3(\mathcal{X})$ to denote the class of three times continuously differentiable functions from $\mathcal{X} \subseteq \mathbb{R}^p$ to \mathbb{R} .

For a given twice differentiable function f such that $\nabla^2 f(x) \succ 0$ at some $x \in \text{dom}(f)$, we define the local norm, and its dual, as

$$\|u\|_x := \langle \nabla^2 f(x) u, u \rangle^{1/2}, \forall u \in \mathbb{R}^p, \quad \text{and} \quad \|v\|_x^* := \max_{\|u\|_x \leq 1} \langle u, v \rangle = \langle \nabla^2 f(x)^{-1} v, v \rangle^{1/2},$$

respectively, for $u, v \in \mathbb{R}^p$. Note that the Cauchy-Schwarz inequality holds, i.e., $\langle u, v \rangle \leq \|u\|_x \|v\|_x^*$.

2.2. Generalized proximity operators. The generalized proximity operator of a proper, closed and convex function g is defined as the following program:

$$\text{prox}_g(u) := \arg \min_{v \in \mathbb{R}^p} \left\{ g(v) + 1/2 \cdot \|v - u\|_x^2 \right\}. \quad (4)$$

When $\nabla^2 f = \mathbb{I}$ —the identity matrix—in the local norm, (4) becomes a standard proximal operator [5]. Computing prox_g might be hard even for such cases. Nevertheless, there exist structured smooth and non-smooth convex functions g with prox_g that comes with a closed-form solution or can be computed with low computational complexity. We capture this idea in the following definition.

DEFINITION 2 (Tractable proximity operator). A proper, closed and convex function $g : \mathbb{R}^p \rightarrow \mathbb{R} \cup \{+\infty\}$ has a *tractable* proximity operator if (4) can be computed efficiently via a closed-form solution or via a polynomial time algorithm.

Examples of such functions include the ℓ_1 -norm—where the proximity operator is the well-known soft-thresholding operator [13]—and the indicator functions of simple sets (*e.g.*, boxes, cones and simplexes)—where the proximity operator is simply the projection operator. Further examples can be found in [5, 13, 40]. Observe that, due to the existence of a closed-form solution for most well-known proximity operators, one can always compute $\text{prox}_{\frac{1}{t}g}$ efficiently and its computational complexity does not depend on the value of the regularization parameter t . Our main result does not require the tractability of computing the proximity operator of g ; it will be used to analyze the overall computational complexity in Subsection 4.7.

Some properties of the proximity operator prox_g are described in the next lemma:

LEMMA 1. *The generalized proximal operator defined in (4) is co-coercive and therefore non-expansive w.r.t. the local norms, i.e.,*

$$[\text{co-coercive}]: \quad (\text{prox}_g(u) - \text{prox}_g(v))^T (u - v) \geq \|\text{prox}_g(u) - \text{prox}_g(v)\|_x^2, \quad (5)$$

$$[\text{nonexpansive}]: \quad \|\text{prox}_g(u) - \text{prox}_g(v)\|_x \leq \|u - v\|_x^*, \quad \forall u, v \in \mathbb{R}^p. \quad (6)$$

The proof to this lemma can be found in [51, Lemma 2].

2.3. Self-concordant functions and self-concordant barriers. A concept used in our analysis is the self-concordance property, introduced by Nesterov and Nemirovskii [33, 37].

DEFINITION 3. A univariate convex function $\varphi \in \mathcal{C}^3(\text{dom}(\varphi))$ is called *standard self-concordant* if $|\varphi'''(\tau)| \leq 2\varphi''(\tau)^{3/2}$ for all $\tau \in \text{dom}(\varphi)$, where $\text{dom}(\varphi)$ is an open set in \mathbb{R} . Moreover, a function $f : \text{dom}(f) \subseteq \mathbb{R}^p \rightarrow \mathbb{R}$ is standard self-concordant if, for any $x \in \text{dom}(f)$ and $v \in \mathbb{R}^p$, the univariate function φ where $\tau \mapsto \varphi(\tau) := f(x + \tau v)$ is standard self-concordant.

DEFINITION 4. A standard self-concordant function $f : \text{dom}(f) \subset \mathbb{R}^p \rightarrow \mathbb{R}$ is a ν -self-concordant barrier for the set $\text{Dom}(f)$ with parameter $\nu > 0$, if

$$\sup_{u \in \mathbb{R}^p} \{2\langle \nabla f(x), u \rangle - \|u\|_x^2\} \leq \nu, \quad \forall x \in \text{dom}(f).$$

In addition, $f(x) \rightarrow \infty$ as x tends to the boundary of $\text{dom}(f)$.

We note that when $\nabla^2 f$ is non-degenerate (particularly, when $\text{dom}(f)$ contains no straight line [33, Theorem 4.1.3.]), a ν -self-concordant function f satisfies

$$\|\nabla f(x)\|_x^* \leq \sqrt{\nu}, \quad \forall x \in \text{dom}(f). \quad (7)$$

Self-concordant functions have non-global Lipschitz gradients and can be used to analyze the complexity of Newton-methods [10, 33, 37], as well as first-order variants [16]. For more details on self-concordant functions and self-concordant barriers, we refer the reader to Chapter 4 of [33].

Several simple sets are equipped with a self-concordant barrier. For instance, $f_{\mathbb{R}_+^p}(x) := -\sum_{i=1}^n \log(x_i)$ is an n -self-concordant barrier of the orthant cone \mathbb{R}_+^p , $f(x, t) = -\log(t^2 - \|x\|_2^2)$ is a 2-self-concordant barrier of the Lorentz cone $\mathcal{L}_{n+1} := \{(x, t) \in \mathbb{R}^p \times \mathbb{R}_+ : \|x\|_2 \leq t\}$, and the semidefinite cone \mathbb{S}_+^n is endowed with the n -self-concordant barrier $f_{\mathbb{S}_+^n}(X) := -\log \det(X)$. In addition, other convex sets, such as hyperbolic polynomials and convex cones, are also characterized by explicit self-concordant barriers [26, 34]. Generally, any closed and convex set—with nonempty interior and not containing a straight line—is endowed with a self-concordant barrier; see [33, 37].

Finally, we define the analytical center \bar{x}_f^* of f as

$$\bar{x}_f^* := \arg \min \{f(x) : x \in \text{int}(\mathcal{X})\} \Leftrightarrow \nabla f(\bar{x}_f^*) = 0. \quad (8)$$

If \mathcal{X} is bounded, then \bar{x}_f^* exists and is unique [35]. Some properties of the analytical center are presented in Section 3. In this paper, we develop algorithms for (1) with a general self-concordant barrier f of \mathcal{X} as defined by Definition 4.

2.4. Basic assumptions. We make the following assumption, regarding problem (1).

ASSUMPTION 1. *The solution set \mathcal{X}^* of (1) is nonempty. The objective function g in (1) is proper, closed and convex, and $\mathcal{X} \cap \text{dom}(g) \neq \emptyset$. The feasible set \mathcal{X} is nonempty, closed and convex (with nonempty interior $\text{int}(\mathcal{X})$) and is endowed with a ν -self-concordant barrier f such that $\text{Dom}(f) := \text{cl}(\text{dom}(f)) = \mathcal{X}$. The analytical center \bar{x}_f^* of f exists.*

Except for the last condition, Assumption 1 is common for interior-point methods. The last condition can be satisfied by adding an auxiliary constraint $\|x\|_2 \leq R$ for sufficiently large R ; this technique has been also used in [37] and it does not affect the solution of (1) when R is large.

3. Re-parameterizing the central path. In this section, we introduce a new parameterization strategy, which will be used in our scheme for (1).

3.1. Barrier formulation and central path of (1). Since \mathcal{X} is endowed with a ν -self-concordant barrier f , according to Assumption A.1, the barrier formulation of (1) is given by

$$F_t^* := \min_{x \in \text{int}(\mathcal{X})} \left\{ F_t(x) := \frac{1}{t} \cdot G(x) + f(x) \equiv \frac{1}{t} \cdot (\langle c, x \rangle + g(x)) + f(x) \right\}, \quad (9)$$

where $t > 0$ is the penalty parameter. We denote by \bar{x}_t^* the solution of (9) at a given value $t > 0$. Define $r_t(x) := \frac{1}{t} \cdot (c + \partial g(x)) + \nabla f(x)$. The optimality condition of (9) is necessary and sufficient for \bar{x}_t^* to be an optimal solution of (9), and can be written as follows:

$$0 \in r_t(\bar{x}_t^*) \equiv \frac{1}{t} \cdot (c + \partial g(\bar{x}_t^*)) + \nabla f(\bar{x}_t^*). \quad (10)$$

We also denote by $\bar{\mathcal{C}} := \{\bar{x}_t^* : t > 0\}$ the set of solutions of (9), which generates a central path (or a solution trajectory) associated with (1). We refer to each solution \bar{x}_t^* as a central point.

3.2. Parameterization of the optimality condition. Let us fix $x^0 \in \text{dom}(f)$; a specific selection of x^0 is provided later on. For given x^0 , let $\xi_0 \in \partial g(x^0)$ be an arbitrary subgradient of g at x^0 , and set $\zeta_0 := \nabla f(x^0) + \frac{1}{t_0} \cdot (c + \xi_0)$. For a given parameter $\eta > 0$, define

$$h_\eta(x) := f(x) - \eta \langle \zeta_0, x \rangle \quad \text{and} \quad r_{t,\eta}(x) := \frac{1}{t} \cdot (c + \partial g(x)) + \nabla h_\eta(x). \quad (11)$$

with the gradient $\nabla h_\eta(x) := \nabla f(x) - \eta \zeta_0$. We further define an η -parameterized version of (9) as

$$H_t^* := \min_{x \in \text{int}(\mathcal{X})} \left\{ H_t(x) := \frac{1}{t} \cdot (\langle c, x \rangle + g(x)) + h_\eta(x) \right\}. \quad (12)$$

We denote by x_t^* the solution of (12), given $t > 0$. Observe that, for a fixed value of $\eta > 0$, the optimality condition of (12) at x_t^* is given by

$$0 \in r_{t,\eta}(x_t^*) \equiv \frac{1}{t} \cdot (c + \partial g(x_t^*)) + \nabla h_\eta(x_t^*). \quad (13)$$

Next, we provide some remarks regarding the η -parameterized problem in (12):

1. Clearly, if we set $\eta = 0$, $h_\eta(x) \equiv f(x)$ and thus, (12) is equivalent to (9). Therefore, for any other value $\eta > 0$, the problem in (12) differs from the original formulation (9) by a factor $-\eta \langle \zeta_0, x \rangle$.
2. Fix parameters $\eta > 0, t > 0$ and let x_t^* be the solution of (12), which is different from the solution \bar{x}_t^* of (9), given the remark above. However, as $t \rightarrow 0$ in a path-following scheme, both x_t^* and \bar{x}_t^* converge to an optimum x^* of (1).
3. Based on the above, for fixed $t > 0$ and different values of η , (12) leads to a family of paths towards x^* of (1).

Our aim in this paper is to properly combine the quantities t_0 , x^0 and η , such that (i) solving iteratively (12) always has fast convergence (even at the initial point x^0) and, (ii) while (12) differs from (9), its solution trajectory is closely related to the solution trajectory of the original barrier formulation. The above are further discussed in the next subsections.

3.3. A functional connection between solutions of (9) and (12). Given the definitions above, let us first study the relationship between *exact* solutions of (9) and (12), for fixed values $t > 0$ and $\eta > 0$.

LEMMA 2. *Let $t > 0$ be fixed. Assume $\eta > 0$ and ζ_0 is chosen such that $m_0 = \eta \|\zeta_0\|_{\bar{x}_t^*}^* < 1$. Define $\bar{\Delta}_t := \|x_t^* - \bar{x}_t^*\|_{\bar{x}_t^*}$ as the local distance between \bar{x}_t^* and x_t^* , the solutions of (9) and (12), respectively. Then,*

$$\bar{\Delta}_t \leq \frac{m_0}{1 - m_0}.$$

Proof. Let x_t^* be the solution of (12) and \bar{x}_t^* be the solution of (9). By the optimality conditions in (10) and (13), we have $-t\nabla f(\bar{x}_t^*) \in \partial G(\bar{x}_t^*)$ and $-t\nabla h_\eta(x_t^*) \in \partial G(x_t^*)$. Moreover, by the convexity of G , we have $\langle \nabla f(\bar{x}_t^*) - \nabla h_\eta(x_t^*), x_t^* - \bar{x}_t^* \rangle \geq 0$. Using the definition $\nabla h_\eta(x) := \nabla f(x) - \eta\zeta_0$, the last inequality leads to

$$\langle \nabla f(x_t^*) - \nabla f(\bar{x}_t^*), x_t^* - \bar{x}_t^* \rangle \leq \eta \langle \zeta_0, x_t^* - \bar{x}_t^* \rangle.$$

Further, by [33, Theorem 4.1.5] and the Cauchy-Schwarz inequality, this inequality implies

$$\frac{\|x_t^* - \bar{x}_t^*\|_{\bar{x}_t^*}}{1 + \|x_t^* - \bar{x}_t^*\|_{\bar{x}_t^*}} \leq \eta \|\zeta_0\|_{\bar{x}_t^*}^* \implies \bar{\Delta}_t \leq \frac{m_0}{1 - m_0}, \quad (14)$$

which completes the proof of this lemma. ■

The above lemma indicates that, while (9) and (12) define different central paths towards x^* , there is an upper bound m_0 on the distance between \bar{x}_t^* and x_t^* , which is controlled by the selection of η, t_0 and x^0 . However, $\|\zeta_0\|_{\bar{x}_t^*}^*$ cannot be evaluated a priori, since \bar{x}_t^* is unknown.

3.4. Estimate upper bound for $\bar{\Delta}_t$. We can overcome this difficulty by using an approximation of the *analytical center point* \bar{x}_f^* in (8). A key property of \bar{x}_f^* is the following [33, Corollary 4.2.1]: Define $n_\nu := \nu + 2\sqrt{\nu}$, where ν is the self-concordant barrier parameter. If f is a logarithmically homogeneous self-concordant barrier, then we set $n_\nu := 1$ [33]. Then, $\|v\|_x^* \leq n_\nu \|v\|_{\bar{x}_f^*}^*$ for any $x \in \text{int}(\mathcal{X})$ and $v \in \mathbb{R}^p$. This observation leads to the following Corollary; the proof easily follows from that of Lemma 2 and the properties above.

COROLLARY 1. *Consider the configuration in Lemma 2 and define $\bar{m}_0 = \eta n_\nu \|\zeta_0\|_{\bar{x}_f^*}^* < 1$. Then,*

$$\bar{\Delta}_t \leq \frac{\bar{m}_0}{1 - \bar{m}_0}, \quad \forall t > 0. \quad (15)$$

Moreover, if we choose the initial point x^0 as $x^0 := \bar{x}_f^*$, then $\bar{m}_0 = \frac{n_\nu \eta}{t_0} \cdot \|c + \xi_0\|_{\bar{x}_f^*}^* \geq m_0$, where m_0 is defined in Lemma 2.

Proof. By [33, Corollary 4.2.1], one observes that $\|\zeta_0\|_{\bar{x}_t^*}^* \leq n_\nu \|\zeta_0\|_{\bar{x}_f^*}^*$, where \bar{x}_f^* is the analytical center of f . Following the same motions with the proof of Lemma 2, we obtain (15). Further, using the property $\nabla f(\bar{x}_f^*) = 0$ and the definition of ζ_0 , we obtain the last statement. ■

In the corollary above, we bound the quantity \bar{m}_0 using the local norm at the analytical center \bar{x}_f^* . This will allow us to estimate the theoretical worst-case bound in Theorem 3, described next. This corollary also suggests to choose an initial point $x^0 := \bar{x}_f^*$, assuming \bar{x}_f^* is available or easy to compute *exactly*. Later in the text, we propose initialization conditions that can be checked *a priori* and, if they are satisfied, such initial points are sufficient for our results to apply. *E.g.*, consider the case where we only approximate \bar{x}_f^* up to a tolerance level (and not exactly computed), where we can decide on the fly whether such approximation is adequate—see Lemma 5 below.

The above observations lead to the following lemma: given a point x , we bound $\|x - \bar{x}_t^*\|_{\bar{x}_t^*}$ by the distance $\|x - x_t^*\|_{x_t^*}$, using the bound (15).

LEMMA 3. Consider the configuration in Corollary 1, such that $\bar{m}_0 < \frac{1}{2}$. Let $\lambda_t(x) := \|x - x_t^*\|_{x_t^*}$ and $\bar{\lambda}_t(x) := \|x - \bar{x}_t^*\|_{\bar{x}_t^*}$, for any $x \in \text{int}(\mathcal{X})$. Then, the following connection between $\lambda_t(x)$ and $\bar{\lambda}_t(x)$ holds:

$$\bar{\lambda}_t(x) \leq \frac{\lambda_t(x)}{1 - \bar{\Delta}_t} + \bar{\Delta}_t \leq \frac{(1 - \bar{m}_0)\lambda_t(x)}{1 - 2\bar{m}_0} + \frac{\bar{m}_0}{1 - \bar{m}_0}. \quad (16)$$

Proof. By definition of the local norm $\bar{\lambda}_t(x)$, we have

$$\begin{aligned} \bar{\lambda}_t(x) &= \langle \nabla^2 f(\bar{x}_t^*)(x - \bar{x}_t^*), x - \bar{x}_t^* \rangle^{1/2} \\ &\leq \langle \nabla^2 f(\bar{x}_t^*)(x_t^* - \bar{x}_t^*), x_t^* - \bar{x}_t^* \rangle^{1/2} + \langle \nabla^2 f(\bar{x}_t^*)(x - x_t^*), x - x_t^* \rangle^{1/2} \\ &\leq \bar{\Delta}_t + (1 - \|x_t^* - \bar{x}_t^*\|_{\bar{x}_t^*})^{-1} \langle \nabla^2 f(x_t^*)(x - x_t^*), x - x_t^* \rangle^{1/2} \\ &= \bar{\Delta}_t + \frac{\lambda_t(x)}{1 - \bar{\Delta}_t}. \end{aligned}$$

Here, in the first inequality, we use the triangle inequality for the weighted norm $\|\cdot\|_{\nabla^2 f(\bar{x}_t^*)}$, while in the second inequality we apply [33, Theorem 4.1.6]. The proof is completed when we use (15) to upper bound the RHS. ■

The above lemma indicates that, given a fixed $t > 0$, any approximate solution to (12) (say \hat{x}_t) that is “good” enough (*i.e.*, the metric $\lambda_t(\hat{x}_t)$ is small), signifies that \hat{x}_t is also “close” to the optimal of (9) (*i.e.*, the metric $\bar{\lambda}_t(\hat{x}_t)$ is bounded by $\lambda_t(\hat{x}_t)$ and, thus, can be controlled). This fact allows the use of (12), instead of (9), and provides freedom to cleverly select initial parameters t_0 and η for faster convergence. The next section proposes such an initialization procedure.

3.5. The choice of initial parameters. Here, we describe how we initialize t_0 and η . Lemma 3 suggests that, for some $\beta \in (0, 1)$, if we can bound $\lambda_t(\cdot) \leq \beta$, then $\bar{\lambda}_t(\cdot)$ is bounded as well as $\bar{\lambda}_t(\cdot) \leq \frac{(1 - \bar{m}_0)\beta}{1 - 2\bar{m}_0} + \frac{\bar{m}_0}{1 - \bar{m}_0}$. This observation leads to the following lemma.

LEMMA 4. Let $\lambda_{t_0}(x^0) := \|x^0 - x_{t_0}^*\|_{x_{t_0}^*}$, where $x_{t_0}^*$ is the solution of (12) at $t := t_0$ and x^0 is an arbitrarily chosen initial point in $\text{dom}(f)$. Let $\xi_0 \in \partial g(x^0)$ and, from (11), $r_{t_0, \eta}(x^0) := \frac{1}{t_0}(c + \xi_0) + \nabla h_\eta(x^0)$. Then, we have

$$\lambda_{t_0}(x^0) \leq \frac{1 - \gamma_{t_0} - \sqrt{1 - 6\gamma_{t_0} + \gamma_{t_0}^2}}{2}, \quad (17)$$

provided that $\gamma_{t_0} := \|r_{t_0, \eta}(x^0)\|_{x^0}^* \equiv |1 - \eta| \|\zeta_0\|_{x^0}^* < 3 - 2\sqrt{2}$ for a particular choice of η .

Proof. Since $x_{t_0}^*$ is the solution of (12) at $t = t_0$, there exists $\xi_{t_0}^* \in \partial g(x_{t_0}^*)$ such that: $\nabla h_\eta(x_{t_0}^*) + \frac{1}{t_0}(c + \xi_{t_0}^*) = 0$. Hence, by the definitions of $r_{t_0, \eta}(x^0)$ and $h_\eta(\eta)$, we obtain

$$-\frac{1}{t_0} \cdot \xi_{t_0}^* = \frac{1}{t_0} \cdot c + \nabla h_\eta(x_{t_0}^*) \quad \Rightarrow \quad \frac{1}{t_0} \cdot (\xi_0 - \xi_{t_0}^*) = r_{t_0, \eta}(x^0) - (\nabla f(x^0) - \nabla f(x_{t_0}^*)).$$

By convexity of g , we have

$$0 \leq \langle \xi_0 - \xi_{t_0}^*, x^0 - x_{t_0}^* \rangle = \langle r_{t_0, \eta}(x^0) - (\nabla f(x^0) - \nabla f(x_{t_0}^*)), x^0 - x_{t_0}^* \rangle.$$

This inequality leads to $\langle \nabla f(x^0) - \nabla f(x_{t_0}^*), x^0 - x_{t_0}^* \rangle \leq \langle r_{t_0, \eta}(x^0), x^0 - x_{t_0}^* \rangle$. Using the self-concordance of f in [33, Theorem 4.1.7] and the Cauchy-Schwarz inequality, we can derive

$$\frac{\lambda_{t_0}(x^0)^2}{1 + \lambda_{t_0}(x^0)} \leq \langle \nabla f(x^0) - \nabla f(x_{t_0}^*), x^0 - x_{t_0}^* \rangle \leq \langle r_{t_0, \eta}(x^0), x^0 - x_{t_0}^* \rangle \leq \|r_{t_0, \eta}(x^0)\|_{x_{t_0}^*}^* \lambda_{t_0}(x^0).$$

Hence, $\frac{\lambda_{t_0}(x^0)}{1+\lambda_{t_0}(x^0)} \leq \|r_{t_0,\eta}(x^0)\|_{x_{t_0}^*}^*$. Moreover, by [33, Theorem 4.1.6], we have $\|r_{t_0,\eta}(x^0)\|_{x_{t_0}^*}^* \leq \frac{\|r_{t_0,\eta}(x^0)\|_{x^0}^*}{1-\lambda_{t_0}(x^0)}$. Combining these two inequalities, we obtain

$$\frac{\lambda_{t_0}(x^0)(1-\lambda_{t_0}(x^0))}{1+\lambda_{t_0}(x^0)} \leq \|r_{t_0,\eta}(x^0)\|_{x^0}^* \equiv \gamma_{t_0}.$$

After few elementary calculations, one can see that if $\gamma_{t_0} < 3 - 2\sqrt{2}$, we obtain (17), which also guarantees its right-hand side of (17) to be positive. ■

In plain words, Lemma 4 provides a recipe for initial selection of parameters: Our goal is to choose an initial point x^0 and the parameters η and t_0 such that $\lambda_{t_0}(x^0) \leq \beta$, for a predefined constant $\beta \in (0, 1)$. The following lemma (i) provides sufficient conditions—that can be checked *a priori*—on the set of initial points that lead to specific η, t_0 configurations and make the results in the previous subsection hold; and (ii) suggests that even an approximation of the analytical center \bar{x}_f^* is sufficient.

LEMMA 5. *The initial point $x^0 \in \text{dom}(f)$ and the parameters η and t_0 need to satisfy the following relations:*

$$\eta \in \left[1 - \frac{\beta(1-\beta)}{(1+\beta)\|\zeta_0\|_{x^0}^*}, 1 + \frac{\beta(1-\beta)}{(1+\beta)\|\zeta_0\|_{x^0}^*} \right] \quad \text{and} \quad \eta < \frac{1-\beta}{(3+\beta)n_\nu\|\zeta_0\|_{\bar{x}_f^*}^*}. \quad (18)$$

If we choose $x^0 \in \text{dom}(f)$ such that

$$\|\nabla f(x^0)\|_{x^0}^* \leq \kappa < \frac{1}{2} \left(2a_0 + 1 - \sqrt{4a_0^2 + 1} \right) \leq \frac{1}{2}, \quad \text{with} \quad a_0 := \frac{1-\beta}{(3+\beta)n_\nu} < \frac{1}{3}, \quad (19)$$

then one can choose $\eta = 1$ and

$$t_0 \geq \frac{\|c + \xi_0\|_{x^0}^*(1-\kappa)(3+\beta)n_\nu}{(1-2\kappa)(1-\beta) - \kappa(1-\kappa)(3+\beta)n_\nu}. \quad (20)$$

In addition, the quantity \bar{m}_0 defined in Corollary 1 is bounded by $\bar{m}_0 \leq \hat{m}_0 := \frac{(1-\kappa)(\kappa+t_0^{-1}\|c+\xi_0\|_{x^0}^*)}{(1-2\kappa)n_\nu}$. As a special case, if $x^0 := \bar{x}_f^*$, then $\nabla f(x^0) = 0$, and we can choose $\eta = 1$ and $t_0 > \frac{(1-\beta)\|c+\xi_0\|_{x^0}^*}{(3+\beta)n_\nu}$. If η is chosen from the first interval in (18), then we can take any $t_0 > \frac{(1-\beta)\|c+\xi_0\|_{x^0}^*}{\eta(3+\beta)n_\nu}$.

Proof. Using (17), we observe that in order to satisfy $\lambda_{t_0}(x^0) \leq \beta$, it is sufficient to require

$$1 - \gamma_{t_0} - \sqrt{1 - 6\gamma_{t_0} + \gamma_{t_0}^2} \leq 2\beta \quad \Rightarrow \quad \gamma_{t_0} \leq \frac{\beta(1-\beta)}{1+\beta}.$$

Since $\gamma_{t_0} = |1-\eta|\|\zeta_0\|_{x^0}^*$, the inequality $\gamma_{t_0} \leq \frac{\beta(1-\beta)}{1+\beta}$ further implies

$$|1-\eta| \leq \frac{\beta(1-\beta)}{(1+\beta)\|\zeta_0\|_{x^0}^*}.$$

Hence, we obtain the first condition of (18).

By our theory and the choice of m_0 as in Lemma 2, it holds $\bar{\lambda}_t(\cdot) \leq \frac{(1-m_0)\beta}{1-2m_0} + \frac{m_0}{1-m_0} < 1$. Since $\frac{m_0}{1-m_0} \leq \frac{m_0}{1-2m_0}$, the last condition can be upper bounded as follows:

$$\frac{(1-m_0)\beta}{1-2m_0} + \frac{m_0}{1-m_0} \leq \frac{(1-m_0)\beta + m_0}{1-2m_0} < 1.$$

This condition suggests to choose η such that $m_0 < \frac{1-\beta}{3+\beta}$. Let $x^0 \in \text{dom}(f)$ be an initial point, then, by Corollary 1, we can enforce $m_0 \leq \bar{m}_0 < \frac{1-\beta}{3+\beta}$. This condition leads to

$$\eta n_\nu \|\zeta_0\|_{\bar{x}_f^*}^* < \frac{1-\beta}{3+\beta},$$

which implies the second condition of (18).

If we take $\eta = 1$, then the second condition of (18) can be written as $\|\zeta_0\|_{\bar{x}_f^*}^* \leq \frac{1-\beta}{(3+\beta)n_\nu}$. Since $\nabla f(\bar{x}_f^*) = 0$, similar to the proof of (14), we can show that $\frac{\|x^0 - \bar{x}_f^*\|_{x^0}}{1 + \|x^0 - \bar{x}_f^*\|_{x^0}} \leq \|\nabla f(x^0)\|_{x^0}^*$. Hence, if $\|\nabla f(x^0)\|_{x^0}^* \leq \kappa < \frac{1}{2}$, we can show that $\|x^0 - \bar{x}_f^*\|_{x^0} \leq \frac{\kappa}{1-\kappa} < 1$. Using this condition, we have

$$\|\zeta_0\|_{\bar{x}_f^*}^* \leq \frac{\|\zeta_0\|_{x^0}^*}{1 - \|x^0 - \bar{x}_f^*\|_{x^0}} \leq \frac{\|\nabla f(x^0)\|_{x^0}^* + t_0^{-1} \|c + \xi_0\|_{x^0}^*}{1 - \|x^0 - \bar{x}_f^*\|_{x^0}} \leq \frac{(1-\kappa)(\kappa + t_0^{-1} \|c + \xi_0\|_{x^0}^*)}{1-2\kappa}.$$

The condition $\|\zeta_0\|_{\bar{x}_f^*}^* \leq \frac{1-\beta}{(3+\beta)n_\nu}$ is guaranteed if we impose

$$t_0^{-1} \|c + \xi_0\|_{x^0}^* \leq \frac{(1-2\kappa)(1-\beta)}{(1-\kappa)(3+\beta)n_\nu} - \kappa = \frac{(1-2\kappa)(1-\beta) - \kappa(1-\kappa)(3+\beta)n_\nu}{(1-\kappa)(3+\beta)n_\nu}. \quad (21)$$

The right hand side is well-defined if $\frac{\kappa(1-\kappa)}{1-2\kappa} < \frac{1-\beta}{(3+\beta)n_\nu} := a_0$, which is equivalent to $0 < \kappa < \frac{1}{2}(2a_0 + 1 - \sqrt{4a_0^2 + 1}) \leq \frac{1}{2}$. This is exactly (19). Next, it is clear that (21) implies (20).

In particular, if we choose as initial point the *analytical center*, i.e., $x^0 = \bar{x}_f^*$, then $\nabla f(x^0) = \nabla f(\bar{x}_f^*) = 0$. By Corollary 1, we can enforce $m_0 \leq \bar{m}_0 < \frac{1-\beta}{3+\beta}$. This condition leads to

$$\eta n_\nu \|\zeta_0\|_{\bar{x}_f^*}^* = \frac{\eta n_\nu}{t_0} \|c + \xi_0\|_{x^0}^* < \frac{1-\beta}{3+\beta}.$$

If we take $\eta = 1$, which satisfies the first condition of (18), then we can choose t_0 such that $t_0 > \frac{(1-\beta)\|c + \xi_0\|_{x^0}^*}{(3+\beta)n_\nu}$. This condition provides a rule to select t_0 as indicated the last statement. ■

4. The single phase proximal path-following algorithm. In this section, we present the main ideas of our algorithm. According to the previous section, to solve (1), one can parameterize the path-following scheme (9) into (12) and, given proper initialization, solve iteratively (12)—i.e., in a path-following fashion, for decreasing values of t .

In the following subsections, we describe schemes to solve (12) up to some accuracy and how the errors, due to approximation, propagate into our theory. Based on these ideas, Subsection 4.3 describes the main recursion of our algorithm, along with the update rule for t parameter. Subsection 4.4 provides a practical stopping criterion procedure, such that an ε -solution is achieved. Subsection 4.5 provides an overview of the algorithm and its theoretical guarantees.

4.1. An exact proximal Newton scheme. In our discussions so far, x_t^* denotes the *exact* solution to (12), for a given value of parameter t . Since finding x_t^* exactly might be infeasible (e.g., there might be no closed-form solution), it is common practice to iteratively solve (12) via first- or second-order Taylor approximations of the smooth part.

In this work, we focus on Newton-type solutions. Let $Q(\cdot; y)$ be the second-order Taylor approximation of $h_\eta(\cdot)$ around y , i.e.,

$$\begin{aligned} Q(x; y) &:= \langle \nabla h_\eta(y), x - y \rangle + \frac{1}{2} \langle \nabla^2 h_\eta(y)(x - y), x - y \rangle \\ &= \langle \nabla f(y) - \eta \zeta_0, x - y \rangle + \frac{1}{2} \langle \nabla^2 f(y)(x - y), x - y \rangle. \end{aligned}$$

Then, x_t^* can be obtained by iteratively solving the following subproblem:

$$x_t^+ \leftarrow \arg \min_{x \in \text{int}(\mathcal{X})} \left\{ \hat{F}_t(x; x_t) := Q(x; x_t) + \frac{1}{t} \cdot G(x) \right\}, \quad (22)$$

where each iteration is computed with *perfect* accuracy. Here, x_t denotes the current estimate and x_t^+ denotes the next estimate. Then, repeating (22) with *infinite* accuracy, we have:

$$x_t^\infty \equiv x_t^*,$$

i.e., we ultimately obtain x_t^* , for fixed $t > 0$. We note that, for given point x_t , we can write the optimality condition of (22) as follows:

$$0 \in \nabla h_\eta(x_t) + \nabla^2 h_\eta(x_t)(x_t^* - x_t) + \frac{1}{t} \cdot (c + \partial g(x_t^*)). \quad (23)$$

To solve (22), one can use composite convex quadratic minimization solvers; see, *e.g.*, [6, 9, 36]. The efficiency of such solvers affects the overall analytical complexity and is discussed later in the text (see Subsection 4.7).

4.2. Inexactness in proximal Newton steps. In practice, we cannot solve (22) exactly, but only hope for an approximate solution, up to a given accuracy $\delta > 0$ [25]. The next definition characterizes such inexact solutions.

DEFINITION 5. Fix $t > 0$ and let w be an anchor point (in (22), $w = x_t$). Moreover, let x_t^* be the exact solution, obtained by solving (22) exactly. We say that a point $z \in \text{int}(\mathcal{X})$ is a δ -solution to (22) if

$$\|z - x_t^*\|_w \leq \delta, \quad (24)$$

for a given tolerance $\delta \geq 0$. We denote this notion by $z \approx x_t^*$.

Nevertheless, x_t^* is unknown and, thus, we cannot check the condition (24). This condition however holds indirectly, when the following holds [50]:

$$\hat{F}_t(z; w) - \hat{F}_t(x_t^*; w) \leq \frac{\delta^2}{2}, \quad (25)$$

where $\hat{F}_t(\cdot; \cdot)$ is defined in (22). This last condition can be evaluated via several convex optimization algorithms, including first-order methods, *e.g.*, [6, 36].

We will use these ideas next to define our inexact proximal-Newton path-following scheme.

4.3. A new, inexact proximal-Newton path-following scheme. Here, we design a new, path-following scheme that operates over the re-parameterized central path in (12). This new algorithm chooses an initial point, as described in Section 3, and selects values for parameter t via a new update rule, that differs from that of [50].

At the heart of our approach lies the following recursive scheme:

$$\begin{cases} t_{k+1} &:= t_k + d_k, \\ x_{t_{k+1}} &\approx \arg \min_{x \in \text{int}(\mathcal{X})} \left\{ \hat{F}_{t_{k+1}}(x; x_{t_k}) := Q(x; x_{t_k}) + \frac{1}{t_{k+1}} \cdot G(x) \right\}. \end{cases} \quad (26)$$

That is, starting from initial points t_0 and $x^0 \equiv x_{t_0}$, we update the penalty parameter t_k to t_{k+1} via the rule $t_{k+1} := t_k + d_k$, at the k -th iteration; see next for details. Then, we perform a *single* proximal-Newton iteration, in order to approximate the solution to the minimization problem in (26). Observe that, while such a step only approximates the minimizer of (26), by satisfying (24) in our analysis, we can still guarantee convergence close to x^* of (1).

Notation	Description
x^*	Optimal solution of (1).
\bar{x}_t^*	Exact solution of (9) for fixed t .
x_t^*	Exact solution of the parametrized problem (12) for fixed t .
$\bar{x}_{t_{k+1}}$	Exact solution of (26) around x_{t_k} for the penalty parameter t_{k+1} .
$\bar{x}_{t_{k+1}}^*$	Inexact solution of (26) around x_{t_k} for the penalty parameter t_{k+1} .
$\bar{\lambda}_t(x)$	$\bar{\lambda}_t(x) := \ x - \bar{x}_t^*\ _{\bar{x}_t^*}$
$\lambda_t(x)$	$\lambda_t(x) := \ x - x_t^*\ _{x_t^*}$.

TABLE 1. Key notation.

Update rule for penalty parameter. We define the local distances

$$\lambda_t(x) := \|x - x_t^*\|_{x_t^*}, \quad \text{and} \quad \Delta_k := \|x_{t_{k+1}}^* - x_{t_k}^*\|_{x_{t_{k+1}}^*}, \quad (27)$$

for any $t > 0$ and $x \in \text{int}(\mathcal{X})$. Before we provide a closed-form formula for d_k in the update rule $t_{k+1} := t_k + d_k$ above, we require the following lemma, which reveals the relationship between $\lambda_{t_{k+1}}(x_{t_{k+1}})$ and $\lambda_{t_{k+1}}(x_{t_k})$, as well as the relationship between $\lambda_{t_k}(x_{t_k})$ and Δ_k .

LEMMA 6. *Given x_{t_k} and t_{k+1} , let $x_{t_{k+1}}$ be an approximation of $x_{t_{k+1}}^*$ computed by the inexact proximal-Newton scheme (26). Let $\lambda_{t_{k+1}}(x_{t_k}) = \|x_{t_k} - x_{t_{k+1}}^*\|_{x_{t_{k+1}}^*}$ and $\lambda_{t_{k+1}}(x_{t_{k+1}}) = \|x_{t_{k+1}} - x_{t_{k+1}}^*\|_{x_{t_{k+1}}^*}$, according to (27). Let $\delta \geq 0$ be the user-defined approximation parameter, according to (24), where $w \equiv x_{t_{k+1}}^*$ at the $(k+1)$ -th iteration. If $\lambda_{t_{k+1}}(x_{t_k}) \in [0, 1/9]$, then we have*

$$\lambda_{t_{k+1}}(x_{t_{k+1}}) \leq \frac{17}{15} \cdot \delta + 5\lambda_{t_{k+1}}(x_{t_k})^2. \quad (28)$$

The right-hand side of (28) is nondecreasing w.r.t. $\lambda_{t_{k+1}}(x_{t_k})$ and $\delta \geq 0$. Furthermore, let Δ_k be defined by (27). Then, we have the following estimate:

$$\lambda_{t_{k+1}}(x_{t_k}) \leq \frac{\lambda_{t_k}(x_{t_k})}{1 - \Delta_k} + \Delta_k, \quad (29)$$

provided that $\Delta_k < 1$.

Proof. It is proved in [50, Theorem 3.3] that

$$\lambda_{t_{k+1}}(x_{t_{k+1}}) \leq \frac{\delta}{1 - \lambda_{t_{k+1}}(x_{t_k})} + \left(\frac{3 - 2\lambda_{t_{k+1}}(x_{t_k})}{1 - 4\lambda_{t_{k+1}}(x_{t_k}) + 2\lambda_{t_{k+1}}(x_{t_k})^2} \right) \lambda_{t_{k+1}}(x_{t_k})^2, \quad (30)$$

where $\lambda_{t_{k+1}}(x_{t_k}) < 1 - \frac{1}{\sqrt{2}}$. To obtain this result, we require the properties of the proximity operator $\text{prox}_g(\cdot)$, according to Lemma 1. Next, we consider the function $m(q) := \frac{3-2q}{1-4q+2q^2}$ for $q \in [0, 1 - \frac{1}{\sqrt{2}}]$, in order to model the second term in the right hand side of (30). We numerically check that if $q \in [0, 1/9]$ then $m(q) \leq 5$. In this case, we also have $\frac{1}{1-q} \leq \frac{17}{15}$. Using these upper bounds into (30), we obtain $\lambda_{t_{k+1}}(x_{t_{k+1}}) \leq \frac{17}{15} \cdot \delta + 5\lambda_{t_{k+1}}(x_{t_k})^2$, whenever $\lambda_{t_{k+1}}(x_{t_k}) \in [0, 1/9]$.

For the (29), we have

$$\begin{aligned} \lambda_{t_{k+1}}(x_{t_k}) &= \langle \nabla^2 f(x_{t_{k+1}}^*)(x_{t_k} - x_{t_{k+1}}^*), x_{t_k} - x_{t_{k+1}}^* \rangle^{1/2} \\ &\leq \langle \nabla^2 f(x_{t_{k+1}}^*)(x_{t_k} - x_{t_{k+1}}^*), x_{t_k} - x_{t_{k+1}}^* \rangle^{1/2} + \langle \nabla^2 f(x_{t_{k+1}}^*)(x_{t_k} - x_{t_k}^*), x_{t_k} - x_{t_k}^* \rangle^{1/2} \\ &\leq \Delta_k + \left(1 - \|x_{t_k}^* - x_{t_{k+1}}^*\|_{x_{t_{k+1}}^*} \right)^{-1} \langle \nabla^2 f(x_{t_k}^*)(x_{t_k} - x_{t_k}^*), x_{t_k} - x_{t_k}^* \rangle^{1/2} \\ &= \Delta_k + \frac{\lambda_{t_k}(x_{t_k})}{1 - \Delta_k}. \end{aligned}$$

Here, in the first inequality, we use the triangle inequality for the weighted norm $\|\cdot\|_{\nabla^2 f(x_{t_{k+1}}^*)}$, while in the second inequality we apply [33, Theorem 4.1.6]. ■

In words, (28) states that the quadratic convergence rate of consecutive inexact proximal-Newton steps in (26) is preserved per iteration. Moreover, (28) describes how the approximation parameter δ accumulates as the iteration counter increases (*i.e.*, it introduces an additive error term).

The next lemma shows how we can bound Δ_k based on the update rule $t_{k+1} = t_k + d_k$ for $d_k \neq 0$. This lemma also provides a rule for d_k selection.

LEMMA 7. *Given constant $c_\beta > 0$, let $\sigma_\beta := \frac{c_\beta}{(1+c_\beta)\sqrt{\nu}}$. Then, Δ_k defined in (27) satisfies*

$$\frac{\Delta_k}{1 + \Delta_k} \leq \frac{|d_k|}{t_k} \|\nabla f(x_{t_{k+1}}^*)\|_{x_{t_{k+1}}^*}^* \leq \frac{|d_k| \sqrt{\nu}}{t_k}. \quad (31)$$

Moreover, if we choose $d_k := -\sigma_\beta t_k$, the Δ_k is bounded by $\Delta_k \leq c_\beta$.

Proof. Since $x_{t_k}^*$ and $x_{t_{k+1}}^*$ are the exact solutions of (12) at $t = t_k$ and t_{k+1} , respectively, we have

$$0 \in \nabla h_\eta(x_{t_k}^*) + \frac{1}{t_k} \cdot \partial G(x_{t_k}^*) \quad \text{and} \quad 0 \in \nabla h_\eta(x_{t_{k+1}}^*) + \frac{1}{t_{k+1}} \cdot \partial G(x_{t_{k+1}}^*).$$

Hence, there exist $v_{t_k}^* \in \partial G(x_{t_k}^*)$ and $v_{t_{k+1}}^* \in \partial G(x_{t_{k+1}}^*)$ such that $v_{t_k}^* = -t_k \nabla h_\eta(x_{t_k}^*)$ and $v_{t_{k+1}}^* = -t_{k+1} \nabla h_\eta(x_{t_{k+1}}^*)$. Using the convexity of G , we have

$$\langle t_{k+1} \nabla h_\eta(x_{t_{k+1}}^*) - t_k \nabla h_\eta(x_{t_k}^*), x_{t_{k+1}}^* - x_{t_k}^* \rangle = -\langle v_{t_{k+1}}^* - v_{t_k}^*, x_{t_{k+1}}^* - x_{t_k}^* \rangle \leq 0. \quad (32)$$

By the definition ∇h_η and the update rule (26) of t_k , we have

$$t_{k+1} \nabla h_\eta(x_{t_{k+1}}^*) - t_k \nabla h_\eta(x_{t_k}^*) = t_k [\nabla f(x_{t_{k+1}}^*) - \nabla f(x_{t_k}^*)] + d_k \nabla f(x_{t_{k+1}}^*). \quad (33)$$

Combining (32) and (33), then using [33, Theorem 4.1.7], the Cauchy-Schwarz inequality, and the definition of Δ_k in (27) we obtain

$$\begin{aligned} \frac{t_k \Delta_k^2}{1 + \Delta_k} &\leq t_k \langle \nabla f(x_{t_{k+1}}^*) - \nabla f(x_{t_k}^*), x_{t_{k+1}}^* - x_{t_k}^* \rangle \\ &\leq -d_k \langle \nabla f(x_{t_{k+1}}^*), x_{t_{k+1}}^* - x_{t_k}^* \rangle \\ &\leq |d_k| \|\nabla f(x_{t_{k+1}}^*)\|_{x_{t_{k+1}}^*}^* \Delta_k, \end{aligned} \quad (34)$$

which implies the first inequality of (31). The second inequality of (31) follows from the fact that $\|\nabla f(x_{t_{k+1}}^*)\|_{x_{t_{k+1}}^*}^* \leq \sqrt{\nu}$ due to [33, formula 2.4.2]. The last statement of this lemma is a direct consequence of (31). ■

Based on the above, we describe next the main result of this section: Assume that the point x_{t_k} is in the β -neighborhood of the inexact proximal-Newton method (26), *i.e.*, $\lambda_{t_k}(x_{t_k}) \leq \beta$ for given $\beta \in (0, 1/9]$, according to Lemma 6. The following theorem describes a condition on Δ_k such that $\lambda_{t_{k+1}}(x_{t_{k+1}}) \leq \beta$. This in sequence determines the update rule of t values. The following theorem summarizes this requirement.

THEOREM 2. *Let $\{\lambda_{t_k}(x_{t_k})\}$ be the sequence generated by the inexact proximal-Newton scheme (26). For any $\beta \in (0, 1/9]$, if we choose δ and Δ_k such that*

$$\delta \leq \frac{\beta}{16} \quad \text{and} \quad \Delta_k \leq \frac{1}{2} \left(1 + 0.43\sqrt{\beta} - \sqrt{(1 - 0.43\sqrt{\beta})^2 + 4\beta} \right), \quad (35)$$

then the condition $\lambda_{t_k}(x_{t_k}) \leq \beta$ implies $\lambda_{t_{k+1}}(x_{t_{k+1}}) \leq \beta$. Consequently, the penalty parameter t_k is updated by

$$t_{k+1} := (1 - \sigma_\beta) t_k = \left(1 - \frac{c_\beta}{(1 + c_\beta)\sqrt{\nu}} \right) t_k, \quad (36)$$

which guarantees that Δ_k satisfies the condition (35), where

$$c_\beta := \frac{1}{2} \left(1 + 0.43\sqrt{\beta} - \sqrt{(1 - 0.43\sqrt{\beta})^2 + 4\beta} \right) \in (0, 1/23].$$

In addition, $c_\beta^{\max} := \max \{c_\beta : \beta \in (0, 1/9]\} \approx \frac{1}{23}$ when $\beta \approx 0.042231$.

Proof. Under the assumption $\lambda_{t_k}(x_{t_k}) \leq \beta$ and the first condition (35) with $\delta \leq \frac{\beta}{16}$, we can obtain from (28) and (29) that $\lambda_{t_{k+1}}(x_{t_{k+1}}) \leq \frac{2\beta}{27} + 5\left(\frac{\beta}{1-\Delta_k} + \Delta_k\right)^2$. To guarantee $\lambda_{t_{k+1}}(x_{t_{k+1}}) \leq \beta$, we have

$$\frac{2\beta}{27} + 5\left(\frac{\beta}{1-\Delta_k} + \Delta_k\right)^2 \leq \beta \Rightarrow \frac{\beta}{1-\Delta_k} + \Delta_k \leq 0.43\sqrt{\beta}.$$

The last condition implies

$$\Delta_k \leq \frac{1}{2} \left(1 + 0.43\sqrt{\beta} - \sqrt{(1 - 0.43\sqrt{\beta})^2 + 4\beta} \right).$$

This is the second condition of (35), provided that $\beta \in (0, 1/9]$. The second statement of this theorem follows from (35) and Lemma 7, while the last statement is computed numerically. ■

4.4. Stopping criterion. We require a stopping criterion that guarantees an ε -solution for (1) according to Definition 1. First, let us define the following quantities

$$\gamma_1 := \frac{(1 - \hat{m}_0)\beta}{1 - 2\hat{m}_0} + \frac{\hat{m}_0}{1 - \hat{m}_0} \quad \text{and} \quad \hat{\gamma}_1 := \frac{0.43\sqrt{\beta}(1 - \hat{m}_0)}{1 - 2\hat{m}_0} + \frac{\hat{m}_0}{1 - \hat{m}_0}, \quad (37)$$

where $\hat{m}_0 := \frac{(1-\kappa)(\kappa+t_0^{-1}\|c+\xi_0\|_{x_0}^*)}{(1-2\kappa)n_\nu}$, as defined in Lemma 5, and $\beta \in (0, 1/9]$. By the proof of Lemma 5, one can show that $\gamma_1 < 1$ and $\hat{\gamma}_0 < 1$. Next, we present the following Lemma that provides a stopping criterion for our algorithm below; the proof is provided in the Appendix 7.

LEMMA 8. *Let $\{x_{t_k}\}$ be the sequence generated by (26). Then, it holds $\forall k$ that $\{x_{t_k}\} \subset \mathcal{X}$ and*

$$0 \leq G(x_{t_{k+1}}) - G^* \leq t_{k+1} \cdot \psi_\beta(\nu), \quad (38)$$

where

$$\psi_\beta(\nu) := \nu + \sqrt{\nu} \frac{\gamma_1}{1 - \hat{\gamma}_0} + \frac{\hat{\gamma}_0}{(1 - \hat{\gamma}_0)^2} (\hat{\gamma}_0 + \gamma_1 + \delta) + \frac{\delta^2}{2} + \hat{m}_0 \gamma_1. \quad (39)$$

I.e., for a given tolerance $\varepsilon > 0$, if $t_{k+1} \cdot \psi_\beta(\nu) \leq \varepsilon$, then $x_{t_{k+1}}$ is an ε -solution of (1), according to Definition 1.

4.5. Overview of our scheme. We summarize the proposed scheme in Algorithm 1. It is clear that the computational bottleneck lies in Step 12, where we approximately solve a strongly convex quadratic composite subproblem. We comment on this step and its solution in Subsection 4.7. The following theorem summarizes the worst-case iteration-complexity of Algorithm 1.

THEOREM 3. *Let $\{(x_{t_k}, t_k)\}$ be the sequence generated by Algorithm 1. Then, the total number of iterations required to reach an ε -solution x of (1) does not exceed*

$$k_{\max} := \left\lceil \frac{\log\left(\frac{\psi_\beta(\nu)}{t_0 \varepsilon}\right)}{-\log(1 - \sigma_\beta)} \right\rceil + 1. \quad (40)$$

Thus, the worst-case iteration-complexity of Algorithm 1 is $\mathcal{O}\left(\sqrt{\nu} \log\left(\frac{\nu}{t_0 \varepsilon}\right)\right)$.

Algorithm 1 Single-phase, proximal path-following scheme

- 1: **Input:** Tolerance $\varepsilon > 0$.
- 2: **Initialization:**
- 3: Choose $\beta \in (0, 1/9]$. Set $a_0 := \frac{(1-\beta)}{(3+\beta)n_\nu}$ and $\delta := \frac{\beta}{16}$.
- 4: Find a point $x^0 \in \text{dom}(f)$ such that $\|\nabla f(x^0)\|_{x^0}^* \leq \kappa$ for some $\kappa \in (0, \frac{1}{2}(2a_0+1-\sqrt{4a_0^2+1}))$.
- 5: Compute a subgradient $\xi_0 \in \partial g(x^0)$ and compute $c_0 := \|c + \xi_0\|_{x^0}^*$.
- 6: Set $\eta := 1$, and choose $t_0 > \frac{(1-\beta)c_0}{(3+\beta)n_\nu}$.
- 7: Compute \hat{m}_0 from Lemma 5, $\hat{\gamma}_0$ and γ_1 from (37), and $\psi_\beta(\nu)$ from (39) of Lemma 8.
- 8: Set $c_\beta := \frac{1}{2}[1 + 0.43\sqrt{\beta} - \sqrt{(1 - 0.43\sqrt{\beta})^2 + 4\beta}]$ and $\sigma_\beta := \frac{c_\beta}{(1+c_\beta)\sqrt{\nu}}$.
- 9: **for** $k := 0$ **to** k_{\max} **do**
- 10: If $t_k \psi_\beta(\nu) \leq \varepsilon$, then TERMINATE.
- 11: Update $t_{k+1} := (1 - \sigma_\beta)t_k$.
- 12: Perform the inexact full-step proximal-Newton iteration by solving

$$x_{t_{k+1}} := \underset{x}{\operatorname{argmin}} \left\{ \hat{F}_{t_{k+1}}(x; x_{t_k}) := Q(x; x_{t_k}) + \frac{1}{t_{k+1}} \cdot G(x) \right\}, \quad \text{up to a given accuracy } \delta.$$

13: **end for**

Proof. From (36), we can see that $t_k = (1 - \sigma_\beta)^k t_0$. Hence, to obtain $0 \leq G(x_{t_{k+1}}) - G^* \leq \varepsilon$, using (10), we require

$$t_k \geq \frac{\psi_\beta(\nu)}{\varepsilon}, \quad \text{or} \quad k \geq \frac{\log\left(\frac{\psi_\beta(\nu)}{t_0 \varepsilon}\right)}{-\log(1 - \sigma_\beta)}.$$

By rounding up this estimate, we obtain k_{\max} as in (40). We note that $-\log(1 - \sigma_\beta) = \mathcal{O}(1/\sqrt{\nu})$. In addition, by (39), we have $\psi_\beta(\nu) = \mathcal{O}(\nu)$. Hence, the worst-case iteration-complexity of Algorithm 1 is $\mathcal{O}\left(\sqrt{\nu} \log\left(\frac{\nu}{t_0 \varepsilon}\right)\right)$. ■

We note that the worst-case iteration-complexity stated in Theorem 3 is a global worst-case complexity, which is different from the one in [50]. As already mentioned in the Introduction, in the latter case we require

$$\left\lceil \frac{F_{t_0}(x^0) - F_{t_0}(x_{t_0}^*)}{\omega((1 - \kappa)\beta)} \right\rceil$$

iterations in PHASE I, for arbitrary selected t_0 and x^0 , and $\kappa \in (0, 1)$, $\beta \in (0, 0.15]$, $\omega(q) = q - \log(1 + q)$, i.e., PHASE I has a sublinear convergence rate to the initial point $x_{t_0}^0$.

We illustrate the basic idea of our single-phase scheme compared to the two-phase scheme in [50] in Figure 1. Our method follows different central path generated by the solution trajectory of the re-parameterized barrier problem, where an initial point x^0 is immediately available.

4.6. The exact variant. We consider a special case of Algorithm 1, where the subproblem (22) at Step 9 can be solved exactly to obtain $x_{t_{k+1}}$ such that $x_{t_{k+1}} = \bar{x}_{t_{k+1}}^{k+1}$. In this case, we can enlarge the constant c_β in (36) to obtain a better factor σ_β . More precisely, we can use

$$\bar{c}_\beta := \frac{1}{2} \left[1 + 0.45\sqrt{\beta} - \sqrt{(1 - 0.45\sqrt{\beta})^2 + 4\beta} \right] > c_\beta, \quad (41)$$

where $\beta \in (0, 0.116764]$. Hence, we obtain a faster convergence (up to a constant factor) in this case. For instance, we can numerically check that $\bar{c}_\beta^{\max} := \max\{\bar{c}_\beta : \beta \in (0, 0.116764]\} = 0.048186 > c_\beta^{\max} = 0.044183$ with respect to $\beta = 0.045864$.

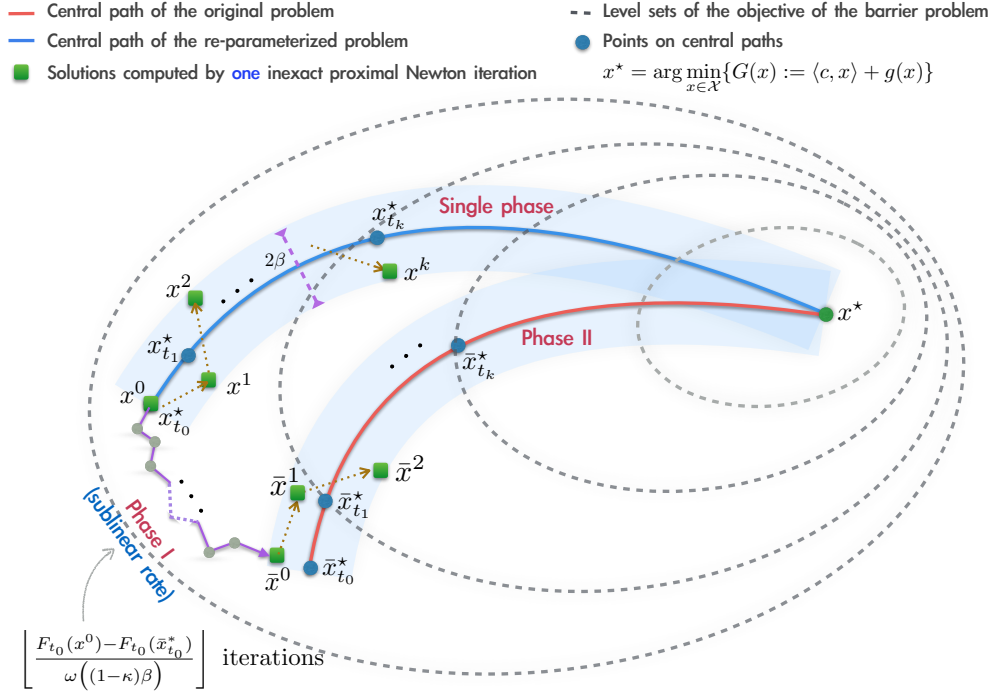


FIGURE 1. Illustration of differences between path-following trajectories followed by single-phase Algorithm 1 and two-phase algorithm in [50]. In the latter case and given an initial point, say $x^0 \equiv \bar{x}_f^*$, [50] first performs $\left\lceil \frac{F_{t_0}(x^0) - F_{t_0}(\bar{x}_{t_0}^*)}{\omega((1-\kappa)\beta)} \right\rceil$ iterations for PHASE I to obtain an initial point \bar{x}^0 , within the quadratic convergence region of Newton method. Then, the fast convergent PHASE II follows the central path (in red color) towards x^* . Our algorithm avoids the sublinearly convergent PHASE I by properly selecting t_0, η and x^0 , and follows a different central path generated by the solution trajectory of the re-parameterized barrier problem (blue curve).

4.7. Overall computational complexity of Algorithm 1 in practice. In this subsection, we present the actual computational complexity of our implementation. Key step in the proposed scheme is the subproblem (22) (or Step 12 in Algorithm 1) and how we can efficiently complete it. Let $\bar{x}_{t_{k+1}}$ be the exact solution of this problem, which exists and is unique due to the strong convexity of (22). Our objective is to solve this problem approximately in order to obtain a δ -approximate solution $x_{t_{k+1}}$, according to (25).

Observe that the subproblem (22) contains the second-order Taylor approximation of $h_\eta(\cdot)$; in particular, it contains the term $Q(x; x_{t_k}) := \langle \nabla f(x_{t_k}) - \eta \zeta_0, x - x_{t_k} \rangle + \frac{1}{2} \langle \nabla^2 f(x_{t_k})(x - x_{t_k}), x - x_{t_k} \rangle$, which is strongly convex. The performance of solvers for (22) depends entirely on the structure of $\nabla^2 f(x_{t_k})$. We consider the following two cases:

- If the solution x^* of (1) does not lie on the boundary of \mathcal{X} due to the nonlinearity of g , then the condition number $\kappa_k := \text{cond}(\nabla^2 f(x_{t_k}))$ can be bounded by a fixed constant, independent of the dimension p and the solution tolerance ε .
- If the solution x^* of (1) lies on the boundary of \mathcal{X} , then similar to the proof in [1], the condition number κ_k can grow with rate $\mathcal{O}\left(\frac{1}{t_k}\right)$. In this case, the worst-case upper bound on κ_k is $\mathcal{O}(\varepsilon^{-1})$ if x_{t_k} is an ε -solution since $t_k \leq \varepsilon$.

We note that, in the second case, the overall computational complexity of Algorithm 1 could grow, due to the condition number κ_k . However, our algorithm operates directly on the original problem (1) with the dimension of p : as we show in the experiments, this is often much better than transforming it into a standard cone problem in a much higher dimensional space.

To approximately solve (22), we can apply methods such as accelerated gradient descent, Frank-Wolfe, coordinate descent, and stochastic gradient descent variants. In the discussion that follows,

let us use the well-known fast proximal-gradient method (FISTA) [6, 36]. Based on the analysis in [6, 36], in order to guarantee a δ -solution $x_{t_{k+1}}$ using FISTA, we require at most

$$j_k^{\max} := \left\lceil \sqrt{\kappa_k} \log \left(\frac{\beta(1 + \kappa_k)}{\delta} \right) \right\rceil + 1 \quad \text{iterations.} \quad (42)$$

The per-iteration complexity is dominated by the following motions:

- Lipschitz constant approximation for $Q(\cdot; \cdot)$: this step requires approximating the largest eigenvalue of $\nabla^2 f(x_{t_k})$, which translates into $\mathcal{O}(p^2)$ computational complexity via a power method.
- Gradient of $Q(\cdot; \cdot)$: this step requires one matrix-vector multiplication and two vector additions, with computational complexity $\mathcal{O}(p^2)$.
- Proximal operator $\text{prox}_{t_{k+1}g}(\cdot)$: this step requires polynomial time operations as stated in Definition 2; denote this complexity cost by $\mathcal{T}_{\text{prox}}$.

Combining the above, the overall per-iteration complexity for approximately solving (22) is

$$\mathcal{O} \left(\sqrt{\kappa_k} \log \left(\frac{\beta(1 + \kappa_k)}{\delta} \right) \max \{ \mathcal{T}_{\text{prox}}, p^2 \} \right). \quad (43)$$

We note that the linear convergence rate can be also achieved with restarting procedures and without requiring a lower bound of the strong convexity parameter in Q_k [14].

Combining the per-iteration worst-case computational complexity (43) and the worst-case iteration-complexity in Theorem 3, we obtain the overall computational complexity of Algorithm 1:

$$\mathcal{O} \left(\sqrt{\max_k \kappa_k} \log \left(\frac{\beta(1 + \max_k \kappa_k)}{\delta} \right) \max \{ \mathcal{T}_{\text{prox}}, p^2 \} \cdot \sqrt{\nu} \log \left(\frac{\nu}{t_0 \varepsilon} \right) \right).$$

5. Numerical experiments. In this section, we first discuss some implementation aspects of Algorithm 1: (i) how one can solve efficiently the subproblem in step 9 of Algorithm 1 and, (ii) how we can compute the analytical center \bar{x}_f^* . In sequence, we illustrate the merits of our approach via three numerical examples, where we compare with state-of-the-art interior-point algorithms.

Inexact proximal-Newton step. The key step of Algorithm 1 is the proximal Newton direction. This corresponds to solving the following strongly convex quadratic composite problem:

$$\min_{d \in \mathbb{R}^p} \{ q(d) := \langle h_k, d \rangle + 1/2 \cdot \langle H_k d, d \rangle + g(x^k + d) \}, \quad (44)$$

where $x^k, h_k \in \mathbb{R}^p$, and H_k is a symmetric positive definite matrix.

There exist efficient first-order and proximal quasi-Newton methods that solve (44); see *e.g.*, [6, 7] for concrete instances of proximal methods, as well as [51, 49] for primal and dual approaches on that matter. The efficiency of such algorithms strongly depends on the computation of prox_g . In addition, since (44) is strongly convex, restart strategies, as in [14, 39, 46] for first order methods, can lead to faster convergence rates in practice. When g is absent, (44) reduces to a positive definite linear system $H_k d = -h_k$, which can be efficiently solved by conjugate gradient schemes or Cholesky methods.

The analytical center point. To obtain the theoretical complexity bound of Theorem 3, we require an approximation of the analytical center \bar{x}_f^* of the barrier function f . While in most cases approximating \bar{x}_f^* require an iterative procedure, there are several practical cases where \bar{x}_f^* can be computed analytically. For example, if $f(x) := -\sum_{i=1}^p \log(1 - x_i^2)$, *i.e.*, f is the barrier of the box set $\mathcal{X} := \{x \in \mathbb{R}^p : -1 \leq x_i \leq 1, i = 1, \dots, p\}$, then $\bar{x}_f^* = \mathbf{0} \in \mathbb{R}^p$. In the case of $f(X) := -\log \det(X) - \log \det(U - X)$ for the set $\mathcal{X} := \{X \in \mathbb{S}_+^p : 0 \preceq X \preceq U\}$, where \mathbb{S}_+^p denotes the set of positive semi-definite matrices in $p \times p$ dimensions, we have $\bar{X}_f^* = 0.5U$, where \bar{X}_f^* denotes the

analytical center in a matrix form. In general cases, \bar{x}_f^* can be computed after a few Newton iterations. More details on computation of \bar{x}_f^* can be found in [33, 37].

Next, we study three numerical examples. We first compare with the two-phase algorithm in [50]; then, we compare Algorithm 1 with some off-the-shelf interior-point solvers such as SDPT3 [48], SeDuMi [45] and Mosek [3].

5.1. The MAX-CUT problem. In this example, we consider the SDP relaxation of the well-known MAX-CUT problem. In particular, consider the following problem:

$$\max_X \left\{ \frac{1}{4} \langle L, X \rangle : X \succeq 0, \text{diag}(X) = \mathbf{e} \right\}, \quad (45)$$

where $X \in \mathbb{S}_+^p$ is a positive semi-definite optimization variable, L is the Laplacian matrix of the corresponding underlying graph of the problem, $\text{diag}(X)$ denotes the diagonal of X and $\mathbf{e} := (1, 1, \dots, 1)^\top \in \mathbb{R}^p$. The purpose of this section is to compare Algorithm 1 with the two-phase algorithm in [50]. We note that in the latter case, the algorithm is also an inexact proximal interior point method, that follows a two-phase procedure.

If we define $c := -\frac{1}{4}L$, $g(X) := \delta_\Omega(X)$, the indicator of the feasible set $\Omega := \{X \in \mathbb{S}_+^p : \text{diag}(X) = \mathbf{e}\}$, then (45) can be reformulated into (1). In this case, the proximal operator of g is just the projection onto the affine subspace Ω , which can be computed in a closed form. Moreover, (22) can be solved in a closed form: it requires only one Cholesky decomposition and two matrix-matrix multiplications.

TABLE 2. Summary of results on the small-sized MAX-CUT problems. Here, $\text{Error} := \|X^k - X_{\text{SDPT3}}^*\|_F / \|X_{\text{SDPT3}}^*\|_F$ and f_{SDPT3}^* denotes the objective value obtained by using IPM solver SDPT3 [48] with high accuracy. For the case of [50], the two quantities in Iters column denote the number of iterations required for PHASE I and PHASE II, respectively. **g05_n.0** is for unweighted graphs with edge probability 0.5; **pm1s_100.0** is for a weighted graph with edge weights chosen uniformly from $\{-1, 0, 1\}$ and density 0.1; **wd09_100.0** is for a 0.1 density ten graph with integer edge weights chosen from $[-10, 10]$; **t2g20.5555** is for each dimension three two-dimensional toroidal grid graphs with gaussian distributed weights and dimension 20×20 ; **t3g7.5555** is for each dimension three three-dimensional toroidal grid graphs with gaussian distributed weights and dimension $7 \times 7 \times 7$. In these two last problems, the adjacency matrix A is normalized by $\sqrt{\max |A_{ij}|}$.

Name	p	f_{SDPT3}^*	[50]				Algorithm 1			
			$f(X)$	Error	Iters	Time[s]	$f(X)$	Error	Iters	Time[s]
g05.60.0	60	-59.00	-58.94	4.35e-03	160/680	0.40	-58.94	4.35e-03	704	0.32
g05.80.0	80	-80.00	-79.92	4.38e-03	292/772	0.63	-79.92	4.39e-03	799	0.48
g05_100.0	100	-100.00	-99.90	4.41e-03	351/877	0.94	-99.90	4.38e-03	910	0.75
pm1s_100.0	100	-52.58	-52.52	3.76e-03	233/1015	1.40	-52.52	3.77e-03	1042	0.85
w09_100.0	100	-80.75	-80.67	4.20e-03	729/968	1.30	-80.67	4.21e-03	996	0.87
t3g7.5555	343	-20620.30	-20616.76	4.45e-03	107/32	2.23	-20599.78	1.99e-03	89	1.30
t2g20.5555	400	-31163.19	-31153.93	1.33e-02	159/99	3.41	-31154.04	1.24e-02	157	2.21

We test both algorithms on seven small-sized MAX-CUT instances generated by Rudy². We also consider four medium-sized instances from the Gset data set³, which were also generated from Rudy. Both algorithms are tested in a Matlab R2015a environment, running on a MacBook Pro Laptop, 2.6GHz Intel Core i7 with 16GB of memory. The initial value of t_0 is set at $t_0 := 0.025$ for both cases. We terminate the execution if $|f(X^k) - f_{\text{SDPT3}}^*| / |f_{\text{SDPT3}}^*| \leq 10^{-3}$, where $f(X) := -\text{trace}(LX)$.

The results are provided in Tables 2-3. Algorithm 1 outperforms [50] in terms of total computational time, while achieving the same, if not better, solution w.r.t. objective value. We observe the

² <http://biqmac.uni-klu.ac.at/biqmaclib>.

³ <http://www.cise.ufl.edu/research/sparse/matrices/Gset/index.html>.

TABLE 3. Summary of results on the medium-sized MAX-CUT problems. Here, $\text{Error} := \|X^k - X_{\text{SDPT3}}^*\|_F / \|X_{\text{SDPT3}}^*\|_F$ and f_{SDPT3}^* denotes the objective value obtained by using IPM solver SDPT3 [48] with high accuracy. For the case of [50], the two quantities in Iters column denote the number of iterations required for PHASE I and PHASE II, respectively. Each problem Gxx is sparse with %1 to %3 upper triangle nonzero, binary entries.

Name (Gxx)	p	f_{SDPT3}^*	[50]				Algorithm 1			
			$f(X)$	Error	Iters	Time[s]	$f(X)$	Error	Iters	Time[s]
G01	800	-12080.12	-12080.12	1.46e-02	149/805	104.48	-12080.13	1.46e-02	569	62.43
G43	1000	-7029.29	-7029.30	2.03e-02	153/1031	208.82	-7029.30	2.03e-02	712	143.00
G22	2000	-14116.01	-14116.03	3.57e-02	215/623	805.99	-14116.06	3.57e-02	561	741.90
G48	3000	-5998.57	-5998.57	1.38e-02	225/2893	8487.08	-5998.59	1.40e-02	1978	7978.35

following trade-off w.r.t. the algorithm in [50]: if we increase the initial value of t_0 in [50], then the number of iterations in Phase I is decreasing, but the number of iterations in Phase II is increasing. We emphasize that both algorithms use the worst-case update rule without any line-search on the step-size as in off-the-shelf solvers.

5.2. The MAX- k -CUT problem. Here, we consider the SDP relaxation of the MAX- k -CUT problem, proposed in [15, eq. (3)]:

$$\max_X \left\{ \frac{k-1}{2k} \langle L, X \rangle : X \succeq 0, \text{diag}(X) = \mathbf{e}, X \geq -\frac{1}{k-1} E_p \right\}. \quad (46)$$

Here, L is the Laplacian matrix of the corresponding graph, $\mathbf{e} := (1, 1, \dots, 1)^T$, and E_p is the $p \times p$ all-ones matrix. Observe that $X \geq Y$ corresponds to entrywise inequality. Similarly to (45), if we define $c := -\frac{(k-1)}{2k} L$ and $g(X) := \delta_\Omega(X)$ with $\Omega := \left\{ X \in \mathbb{S}_+^p : \text{diag}(X) = \mathbf{e}, X \geq -\frac{1}{k-1} E_p \right\}$, (46) is a special instance of the class of problems described by (1).

We compare Algorithm 1 with three well-established, off-the-shelf interior-point solvers: SDPT3 [48], SeDuMi [45]⁴, and Mosek [3]⁵. We consider synthetically generated p -node graphs, where each edge is generated from a Bern($1/4, 3/4$) probability distribution; we also set $k = 4$. The parameters of Algorithm 1 are set as in the previous example, and all algorithms are terminated if $|f(X^k) - f^*|/|f^*| \leq 10^{-5}$, where f^* is the best optimal value produced by three off-the-shelf solvers. We solve (22) with a fast projected gradient method, with adaptive restart and a warm-start strategy [46]: Such configuration requires few iterations to achieve our desired accuracy.

TABLE 4. Comparison results on the MAX- k -CUT problem. Here, **Iters** is the number of iterations; Time[s] is the computational time in second; $f(X) = -\text{trace}(LX)$; **svars** is the number of slack variables; and **cnstr** is the number of linear constraints. In addition, we have $p(p+1)/2$ variables in X and one SDP constraint.

Size	Lifting		Algorithm 1		SeDuMi		SDPT3		Mosek	
p	svars	cnstr	$f(X)$	Time[s]	$f(X)$	Time[s]	$f(X)$	Time[s]	$f(X)$	Time[s]
50	1,225	1,275	-87.733	7.32	-86.174	4.76	-86.160	2.02	-86.138	3.84
75	2,775	2,850	-166.237	9.80	-166.236	55.41	-166.214	10.76	-166.214	8.91
100	4,950	5,050	-316.741	18.37	-316.746	732.16	-316.709	48.67	-316.653	26.63
150	11,175	11,325	-654.703	73.63	-654.684	5,121.34	-654.539	484.46	-654.673	366.36
200	19,900	20,100	-1185.784	169.08	-1185.783	25,521.39	-1185.760	2,122.91	-1185.647	2,048.95

Table 4 contains some experimental results. Observe that, if p is small, all algorithms perform well, with the off-the-shelf solvers returning faster a good solution. However, when p increases,

⁴ Both implementations include Matlab and optimized C-coded parts.

⁵ Available for academic use at <https://mosek.com>.

their computational time significantly increases, as compared to Algorithm 1. One reason that this happens is that standard SDP solvers require $p(p+1)/2$ slack variables and $p(p+1)/2$ additional linear constraints, in order to process the component-wise inequality constraints. Such reformulation of the problem significantly increases variable and constraint size and, hence, lead to slower execution. In stark contrast, Algorithm 1 handles both linear and inequality constraints by a simple projection, which requires only $p(p+1)/2$ basic operations (flops). Figure 2 graphically illustrates the scalability of the four algorithms under comparison, based on the results contained in Table 4.

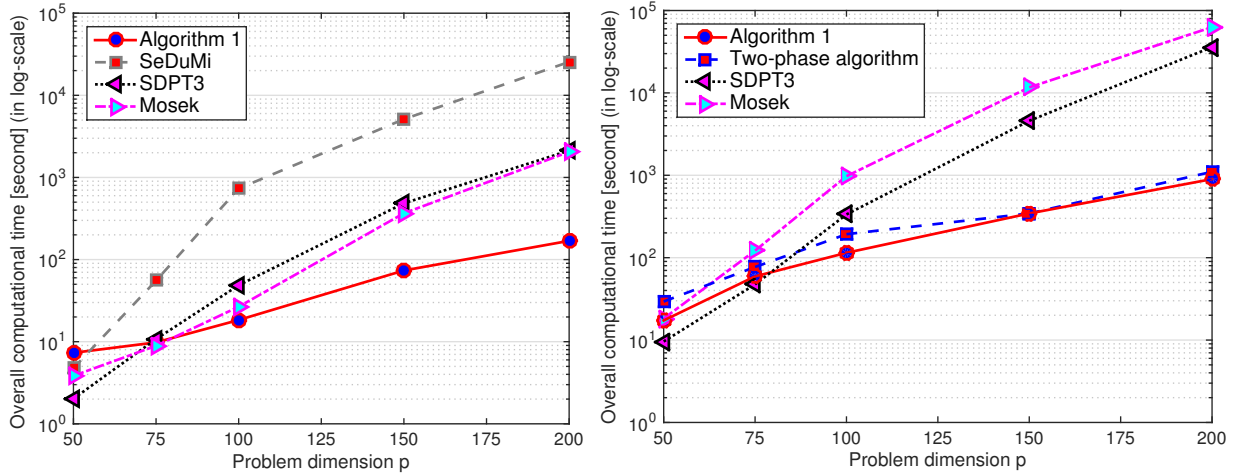


FIGURE 2. Overall execution time, as a function of problem dimension. **Left panel:** MAX- k -CUT problem (46); **Right panel:** Clustering problem (47).

5.3. Max-norm clustering. We consider the max-norm clustering task [21], where we seek a clustering matrix K that minimizes the disagreement with a given affinity matrix A :

$$\begin{aligned} \min_{x := [L, R, K] \in \mathbb{R}^{p \times 3p}} \quad & \|\text{vec}(K - A)\|_1 \\ \text{s.t.} \quad & \mathcal{Q}(x) := \begin{bmatrix} L & K \\ K^T & R \end{bmatrix} \succeq 0, \quad L_{ii} \leq 1, \quad R_{ii} \leq 1, \quad i = 1, \dots, p. \end{aligned} \quad (47)$$

Here, vec is the vectorization operator of a matrix (*i.e.*, $\text{vec}(X) := (X_1^T, \dots, X_n^T)^T$, where X_i is the i -th column of X). Note that (47) is an SDP convex relaxation to the *correlation clustering* problem; see [21] for details. While (47) comes with rigorous theoretical guarantees and can be formulated as a standard conic program, we need to add $O(p^2)$ slack variables to process the ℓ_1 -norm term and the linear constraints. Moreover, the scaling factors (*e.g.*, the Nesterov-Todd scaling factor regarding the semidefinite cone [38]) can create memory bottlenecks in practice, by destroying the sparsity of the underlying problem (*e.g.*, by leading to dense KKT matrices in the Newton systems).

Here, we solve (47) using our path-following scheme. In particular, by defining $x := \text{vec}([K, L, R])$, $f(x) := -\log \det(\mathcal{Q}(x))$ and $g(x) := \|\text{vec}(K - A)\|_1 + \delta_{\Omega}(x)$, we can transform (47) into (1), where δ_{Ω} is the indicator function of $\Omega := \{x : L_{ii} \leq 1, \quad R_{ii} \leq 1, \quad i = 1, \dots, p\}$.

We compare the following solvers: Algorithm 1, the two-phase algorithm in [50], SDPT3 and Mosek. The initial penalty parameter t_0 is set to $t_0 := 0.25$ and the relative tolerance is fixed at 10^{-4} for all algorithms. The data is generated as suggested in [21, 50]. The results of 5 test problem instances are shown in Table 5 sizes p ranging from 50 to 200.⁶

⁶ Since SDPT3 and Mosek cannot run for bigger problems in our personal computer, we restrict to problem sizes up to $p = 200$.

TABLE 5. The performance of Algorithm 1, as compared to three methods on the clustering problem (47). Here, Time[s] is the computational time in second; $f(X) = \|\text{vec}(K - A)\|_1$, and $s\%$ is the sparsity of $K - A$.

Size	Algorithm 1			[50]			SDPT3		Mosek	
p	$f(X)$	Time[s]	$s\%$	$f(X)$	Time[s]	$s\%$	$f(X)$	Time[s]	$f(X)$	Time[s]
50	563.90	17.30	49%	563.90	29.38	49.5%	563.86	9.60	563.92	18.27
75	1,308.19	59.30	43.8%	1,308.18	77.05	43.9%	1,308.15	47.40	1,308.32	121.74
100	2,228.62	114.59	35.8%	2,228.61	192.79	35.9%	2,228.59	334.76	2,228.78	975.10
150	5,328.12	344.29	42.4%	5,327.99	344.32	42.5%	5,327.84	4,584.03	5,328.14	11,665.52
200	9,883.92	899.10	45.8%	9,883.81	1,102.97	47.9%	9,883.68	35,974.60	9,884.21	62,835.42

Both SDPT3 and Mosek are approximately 40 and 60 times slower than Algorithm 1 and [50], especially when $p > 100$. We note that such solvers require $p^2 + 2p$ slack variables, p^2 additional second order cone constraints and $2p$ additional linear constraints to reformulate (47) into a standard SDP problem. Hence, the size of the resulting SDP problem is much larger than of the original one in (47). As a concrete example, if $p = 200$, then this standard mixed cone problem has a (200×200) -SDP variable X , 40,000 second order cone variables, 400 linear variables, 160,400 linear constraints, and 40,000 second order cone constraints. We also see that Algorithm 1 is faster than the two-phase algorithm, in terms of total execution time.

We note that SDPT3 gives a slightly better objective value than Algorithm 1. However, its solution K is fully dense, in contrast to those of Algorithm 1 and [50], reducing its interpretation in applications. Figure 2 (right) also reveals the scalability of these four algorithms for solving (47).

6. Conclusions. We propose a new path-following framework for a, possibly non-smooth, constrained convex minimization template, which includes linear programming as a special case. For our framework, we assume that the constraint set is endowed with a self-concordant barrier and, the non-smooth term has a tractable proximity operator. Our workhorse is a new re-parameterization of the optimality condition of the convex optimization problem, which allows us to select a different central path towards x^* , without relying on the sublinear convergent PHASE I of proximal path-following approaches, as in [50].

We illustrate that the new scheme retains the same global, worst-case, iteration-complexity with standard approaches [33, 37] for smooth convex programming. We theoretically show that inexact solutions to sub-problems do not sacrifice the worst-case complexity, when controlled appropriately. Finally, we numerically illustrate the effectiveness of our framework on MAX-CUT and clustering problems, where the proximal operator play a key role in space efficient optimization.

Acknowledgments: We would like to thank Yurii Nesterov for useful discussions on the initialization technique used in this work. We also thank Dr. Cong Bang Vu for his overall suggestions on the final version of this manuscript. This work was supported in part by NSF grant, no. DMS-16-2044, ERC Future Proof, SNF 200021-146750 and SNF CRSII2-147633.

7. Appendix: The proof of Lemma 8. For notation, see Table 1. By (25) and given $\hat{F}_{t_{k+1}}(x_{t_{k+1}}^{k+1}; x_{t_k}^k) - \hat{F}_{t_{k+1}}(\bar{x}_{t_{k+1}}^{k+1}; x_{t_k}^k) \leq \frac{\delta^2}{2}$, we have

$$\begin{aligned}
 G(x_{t_{k+1}}^{k+1}) &\leq G(\bar{x}_{t_{k+1}}^{k+1}) + t_{k+1}Q_k(\bar{x}_{t_{k+1}}^{k+1}; x_{t_k}^k) - t_{k+1}Q_k(x_{t_{k+1}}^{k+1}; x_{t_k}^k) + \frac{t_{k+1}\delta^2}{2} \\
 &= G(\bar{x}_{t_{k+1}}^{k+1}) + t_{k+1}\langle \nabla f(x_{t_k}^k), \bar{x}_{t_{k+1}}^{k+1} - x_{t_{k+1}}^{k+1} \rangle - t_{k+1}\eta\langle \zeta_0, \bar{x}_{t_{k+1}}^{k+1} - x_{t_{k+1}}^{k+1} \rangle \\
 &\quad + \frac{t_{k+1}}{2} \left(\|\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k\|_{x_{t_k}^k}^2 - \|x_{t_{k+1}}^{k+1} - x_{t_k}^k\|_{x_{t_k}^k}^2 \right) + \frac{t_{k+1}\delta^2}{2}. \tag{48}
 \end{aligned}$$

Now, since $\bar{x}_{t_{k+1}}^{k+1}$ is the exact solution of (22), there exists $\bar{v}^{k+1} \in \partial G(\bar{x}_{t_{k+1}}^{k+1})$ such that

$$\bar{v}^{k+1} = -t_{k+1}(\nabla f(x_{t_k}^k) - \eta\zeta_0) - t_{k+1}\nabla^2 f(x_{t_k}^k)(\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k).$$

Next, using the convexity of G , with $\bar{v}^{k+1} \in \partial G(\bar{x}_{t_{k+1}}^{k+1})$, we have

$$\begin{aligned} G(\bar{x}_{t_{k+1}}^*) - G(\bar{x}_{t_{k+1}}^{k+1}) &\geq \langle \bar{v}^{k+1}, \bar{x}_{t_{k+1}}^* - \bar{x}_{t_{k+1}}^{k+1} \rangle \\ &= -t_{k+1} \langle \nabla f(x_{t_k}^k), \bar{x}_{t_{k+1}}^* - \bar{x}_{t_{k+1}}^{k+1} \rangle + t_{k+1} \eta_0 \langle \zeta_0, \bar{x}_{t_{k+1}}^* - \bar{x}_{t_{k+1}}^{k+1} \rangle \\ &\quad - t_{k+1} \langle \nabla^2 f(x_{t_k}^k)(\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k), \bar{x}_{t_{k+1}}^* - \bar{x}_{t_{k+1}}^{k+1} \rangle \end{aligned} \quad (49)$$

Summing up (48) and (49), and rearranging the terms, we can derive

$$\begin{aligned} G(\bar{x}_{t_{k+1}}^*) - G(x_{t_{k+1}}^{k+1}) &\geq -t_{k+1} \langle \nabla f(x_{t_k}^k), \bar{x}_{t_{k+1}}^* - \bar{x}_{t_{k+1}}^{k+1} \rangle + t_{k+1} \eta \langle \zeta_0, \bar{x}_{t_{k+1}}^* - \bar{x}_{t_{k+1}}^{k+1} \rangle \\ &\quad - t_{k+1} \langle \nabla^2 f(x_{t_k}^k)(\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k), \bar{x}_{t_{k+1}}^* - \bar{x}_{t_{k+1}}^{k+1} \rangle \\ &\quad - \frac{t_{k+1}}{2} \left(\|\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k\|_{x_{t_k}^k}^2 - \|x_{t_{k+1}}^{k+1} - x_{t_k}^k\|_{x_{t_k}^k}^2 \right) - \frac{t_{k+1} \delta^2}{2} \\ &\quad - t_{k+1} \langle \nabla f(x_{t_k}^k), \bar{x}_{t_{k+1}}^{k+1} - x_{t_{k+1}}^{k+1} \rangle + t_{k+1} \eta \langle \zeta_0, \bar{x}_{t_{k+1}}^{k+1} - x_{t_{k+1}}^{k+1} \rangle \\ &= -t_{k+1} \langle \nabla f(x_{t_k}^k), \bar{x}_{t_{k+1}}^* - x_{t_{k+1}}^{k+1} \rangle + t_{k+1} \eta \langle \zeta_0, \bar{x}_{t_{k+1}}^* - x_{t_{k+1}}^{k+1} \rangle \\ &\quad - t_{k+1} \langle \nabla^2 f(x_{t_k}^k)(\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k), \bar{x}_{t_{k+1}}^* - \bar{x}_{t_{k+1}}^{k+1} \rangle \\ &\quad - \frac{t_{k+1}}{2} \left(\|\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k\|_{x_{t_k}^k}^2 - \|x_{t_{k+1}}^{k+1} - x_{t_k}^k\|_{x_{t_k}^k}^2 \right) - \frac{t_{k+1} \delta^2}{2}. \end{aligned} \quad (50)$$

Now, by the Cauchy-Schwarz inequality, we can further estimate (50) as

$$\begin{aligned} G(\bar{x}_{t_{k+1}}^*) - G(x_{t_{k+1}}^{k+1}) &\geq -t_{k+1} \|\nabla f(x_{t_k}^k)\|_{\bar{x}_{t_{k+1}}^*}^* \|\bar{x}_{t_{k+1}}^* - x_{t_{k+1}}^{k+1}\|_{\bar{x}_{t_{k+1}}^*} \\ &\quad - t_{k+1} |\eta| \|\zeta_0\|_{\bar{x}_{t_{k+1}}^*}^* \|\bar{x}_{t_{k+1}}^* - x_{t_{k+1}}^{k+1}\|_{\bar{x}_{t_{k+1}}^*} \\ &\quad - t_{k+1} \langle \nabla^2 f(x_{t_k}^k)(\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k), \bar{x}_{t_{k+1}}^* - \bar{x}_{t_{k+1}}^{k+1} \rangle \\ &\quad - \frac{t_{k+1}}{2} \left(\|\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k\|_{x_{t_k}^k}^2 - \|x_{t_{k+1}}^{k+1} - x_{t_k}^k\|_{x_{t_k}^k}^2 \right) - \frac{t_{k+1} \delta^2}{2}. \end{aligned} \quad (51)$$

We consider the term

$$\mathcal{T}_{[1]} := \|\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k\|_{x_{t_k}^k}^2 - \|x_{t_{k+1}}^{k+1} - x_{t_k}^k\|_{x_{t_k}^k}^2 + 2 \langle \nabla^2 f(x_{t_k}^k)(\bar{x}_{t_{k+1}}^{k+1} - x_{t_k}^k), \bar{x}_{t_{k+1}}^* - \bar{x}_{t_{k+1}}^{k+1} \rangle.$$

Similarly to the proof of [50, Lemma 5.1], we can show that

$$\mathcal{T}_{[1]} \leq \frac{2\bar{\lambda}_{t_{k+1}}(x_{t_k}^k)}{(1 - \bar{\lambda}_{t_{k+1}}(x_{t_k}^k))^2} \left(\bar{\lambda}_{t_{k+1}}(x_{t_k}^k) + \bar{\lambda}_{t_{k+1}}(x_{t_{k+1}}^{k+1}) + \delta \right).$$

Next, by using the self-concordance of f and the definition of $\lambda_t(x)$, we have

$$(1 - \bar{\lambda}_{t_{k+1}}(x_{t_k}^k))^2 \nabla^2 f(\bar{x}_{t_{k+1}}^*) \preceq \nabla^2 f(x_{t_k}^k) \preceq (1 - \bar{\lambda}_{t_{k+1}}(x_{t_k}^k))^{-2} \nabla^2 f(\bar{x}_{t_{k+1}}^*). \quad (52)$$

On the one hand, using (52) and $\|\nabla f(x_{t_k}^k)\|_{x_{t_k}^k}^* \leq \sqrt{\nu}$, we easily get

$$\|\nabla f(x_{t_k}^k)\|_{\bar{x}_{t_{k+1}}^*}^* \leq (1 - \bar{\lambda}_{t_{k+1}}(x_{t_k}^k))^{-1} \|\nabla f(x_{t_k}^k)\|_{x_{t_k}^k}^* \leq (1 - \bar{\lambda}_{t_{k+1}}(x_{t_k}^k))^{-1} \sqrt{\nu}. \quad (53)$$

On the other hand, using \bar{m}_0 defined in Corollary 1, we can show that

$$\|\zeta_0\|_{\bar{x}_{t_{k+1}}^*}^* \stackrel{[33, \text{Corollary 4.1.7}]}{\leq} n_\nu \|\zeta_0\|_{x_f^*}^* \stackrel{\text{Lemma 5}}{=} \bar{m}_0 \leq \hat{m}_0. \quad (54)$$

Substituting (52), (53) and (54) into (51), we finally obtain

$$G(\bar{x}_{t_{k+1}}^*) - G(x_{t_{k+1}}^{k+1}) \geq -t_{k+1} \left(\frac{\sqrt{\nu} \bar{\lambda}_{t_{k+1}}(x_{t_{k+1}}^{k+1})}{(1 - \bar{\lambda}_{t_{k+1}}(x_{t_k}^k))^2} + \eta \bar{m}_0 \bar{\lambda}_{t_{k+1}}(x_{t_{k+1}}^{k+1}) \right. \\ \left. + \frac{\bar{\lambda}_{t_{k+1}}(x_{t_k}^k)}{(1 - \bar{\lambda}_{t_{k+1}}(x_{t_k}^k))^2} \left(\bar{\lambda}_{t_{k+1}}(x_{t_k}^k) + \bar{\lambda}_{t_{k+1}}(x_{t_{k+1}}^{k+1}) + \delta \right) + \frac{\delta^2}{2} \right).$$

Now, let us define the following function

$$\psi(\nu, m_0, \hat{\lambda}_0, \lambda_1, \delta) := \nu + \sqrt{\nu} \frac{\lambda_1}{1 - \hat{\lambda}_0} + \frac{\hat{\lambda}_0}{(1 - \hat{\lambda}_0)^2} \left(\hat{\lambda}_0 + \lambda_1 + \delta \right) + \frac{\delta^2}{2} + m_0 \lambda_1. \quad (55)$$

Using this definition, and combining $G^* - G(\bar{x}_{t_{k+1}}^*) \geq -\nu t_{k+1}$ and the last estimate, we obtain

$$0 \leq G(x_{t_{k+1}}) - G^* \leq t_{k+1} \cdot \psi(\nu, \eta \hat{m}_0, \bar{\lambda}_{t_{k+1}}(x_{t_k}^k), \bar{\lambda}_{t_{k+1}}(x_{t_{k+1}}^{k+1}), \delta). \quad (56)$$

Since $\{(x_{t_k}^k, t_k)\}$ is generated by (26) for $\beta \in (0, 1/9]$, we have $\lambda_{t_{k+1}}(x_{t_{k+1}}) \leq \beta$ and $\lambda_{t_{k+1}}(x_{t_k}^k) \leq 0.43\sqrt{\beta}$. Using Lemma 3, we have

$$\bar{\lambda}_{t_{k+1}}(x_{t_{k+1}}) \leq \frac{(1 - m_0)\beta}{1 - 2m_0} + \frac{m_0}{1 - m_0}, \quad \text{and} \quad \bar{\lambda}_{t_{k+1}}(x_{t_k}^k) \leq \frac{0.43\sqrt{\beta}(1 - m_0)}{1 - 2m_0} + \frac{m_0}{1 - m_0}.$$

From Lemma 5, we see that $m_0 \leq \hat{m}_0 := \frac{(1-\kappa)(\kappa+t_0^{-1}\|c+\xi_0\|_{x_0}^*)}{(1-2\kappa)n_\nu}$, we can show that

$$\bar{\lambda}_{t_{k+1}}(x_{t_{k+1}}) \leq \frac{(1 - \hat{m}_0)\beta}{1 - 2\hat{m}_0} + \frac{\hat{m}_0}{1 - \hat{m}_0} := \gamma_1 \quad \text{and} \quad \bar{\lambda}_{t_{k+1}}(x_{t_k}^k) \leq \frac{0.43\sqrt{\beta}(1 - \hat{m}_0)}{1 - 2\hat{m}_0} + \frac{\hat{m}_0}{1 - \hat{m}_0} := \hat{\gamma}_0.$$

Since $\eta := 1$, using the function ψ defined by (55), if we denote $\psi_\beta(\nu) := \psi(\nu, \hat{m}_0, \hat{\gamma}_0, \gamma_1, \bar{\delta})$, then

$$0 \leq G(x_{t_{k+1}}) - G^* \leq t_{k+1} \cdot \psi_\beta(\nu),$$

which is exactly (38). Using this estimate, if $t_{k+1} \cdot \psi_\beta(\nu) \leq \varepsilon$, then we can say that $x_{t_{k+1}}$ is an ε -solution of (1) in the sense of Definition 1. ■

References

- [1] F. Alizadeh, J.-P. Haeberly, and M. Overton. Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. *SIAM Journal on Optimization*, 8(3):746–768, 1998.
- [2] M. S. Andersen, J. Dahl, Z. Liu, and L. Vandenbergh. Interior-point methods for large-scale cone programming. In: *S. Sra, S. Nowozin, S. J. Wright (ed.) Optimization for Machine Learning*, MIT Press:55–83, 2011.
- [3] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28)*., 2015.
- [4] L. Baldassarre, N. Bhan, V. Cevher, A. Kyrillidis, and S. Satpathi. Group-sparse model selection: Hardness and relaxations. *arXiv preprint arXiv:1303.3207*, 2013.
- [5] H.H. Bauschke and P. Combettes. *Convex analysis and monotone operators theory in Hilbert spaces*. Springer-Verlag, 2011.
- [6] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.
- [7] S. Becker, E. J. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Math. Program. Comput.*, 3(3):165–218, 2011.

- [8] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization: Analysis, algorithms, and engineering applications*, volume 3 of *MPS/SIAM Series on Optimization*. SIAM, 2001.
- [9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- [10] S. Boyd and L. Vandenberghe. *Convex Optimization*. University Press, Cambridge, 2004.
- [11] J.-F. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
- [12] E. Candes, Y. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. *SIAM Journal on Imaging Sciences*, 6(1):199–225, 2013.
- [13] P. Combettes and Pesquet J.-C. Signal recovery by proximal forward-backward splitting. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, 2011.
- [14] O. Fercoq and Z. Qu. Restarting accelerated gradient methods with a rough strong convexity estimate. *arXiv preprint arXiv:1609.07358*, 2016.
- [15] A. Frieze and M. Jerrum. Improved approximation algorithms for maxk-cut and max bisection. *Algorithmica*, 18(1):67–81, 1997.
- [16] R. Frostig, R. Ge, S. Kakade, and A. Sidford. Competing with the empirical risk minimizer in a single pass. *arXiv preprint arXiv:1412.6606*, 2014.
- [17] J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.
- [18] A. Gramfort and M. Kowalski. Improving M/EEG source localization with an inter-condition sparse prior. In *IEEE International Symposium on Biomedical Imaging*, 2009.
- [19] M. Grant, S. Boyd, and Y. Ye. Disciplined convex programming. In L. Liberti and N. Maculan, editors, *Global Optimization: From Theory to Implementation*, Nonconvex Optimization and its Applications, pages 155–210. Springer, 2006.
- [20] K. Jaganathan, S. Oymak, and B. Hassibi. Sparse phase retrieval: Uniqueness guarantees and recovery algorithms. *arXiv preprint arXiv:1311.2745*, 2013.
- [21] A. Jalali and N. Srebro. Clustering using max-norm constrained optimization. In *Proc. of International Conference on Machine Learning (ICML2012)*, pages 1–17, 2012.
- [22] R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research*, 12:2777–2824, 2011.
- [23] R. Jenatton, A. Gramfort, V. Michel, G. Obozinski, F. Bach, and B. Thirion. Multi-scale mining of fmri data with hierarchical structured sparsity. In *Pattern Recognition in NeuroImaging (PRNI)*, 2011.
- [24] A. Kyrillidis, L. Baldassarre, M. El Halabi, Q. Tran-Dinh, and V. Cevher. Structured sparsity: Discrete and convex approaches. In *Compressed Sensing and its Applications*, pages 341–387. Springer, 2015.
- [25] A. Kyrillidis, R. Karimi Mahabadi, Q. Tran Dinh, and V. Cevher. Scalable sparse covariance estimation via self-concordance. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
- [26] A. Lewis and H. Sendov. Self-concordant barriers for hyperbolic means. *Mathematical programming*, 91(1):1–10, 2001.
- [27] Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2009.
- [28] J. Löfberg. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [29] R. Millane. Phase retrieval in crystallography and optics. *JOSA A*, 7(3):394–411, 1990.
- [30] A. Nemirovski. Lectures on modern convex optimization. In *Society for Industrial and Applied Mathematics (SIAM)*. Citeseer, 2001.

- [31] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.
- [32] A. Nemirovski and M. J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17(1):191–234, 2008.
- [33] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.
- [34] Y. Nesterov. Constructing self-concordant barriers for convex cones. Core discussion paper, 2006/30, UCL, CORE Discussion paper, 2006/30, 2006.
- [35] Y. Nesterov. Barrier subgradient method. *Math. Program., Ser. B*, 127:31–56, 2011.
- [36] Y. Nesterov. Gradient methods for minimizing composite objective function. *Math. Program.*, 140(1):125–161, 2013.
- [37] Y. Nesterov and A. Nemirovski. *Interior-point Polynomial Algorithms in Convex Programming*. Society for Industrial Mathematics, 1994.
- [38] Y. Nesterov and M.J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Research*, 22(1):1–42, 1997.
- [39] B. O’Donoghue and E. Candes. Adaptive Restart for Accelerated Gradient Schemes. *Found. Comput. Math.*, 15:715–732, April 2015.
- [40] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.
- [41] F. Rapaport, E. Barillot, and J.P. Vert. Classification of arrayCGH data using fused SVM. *Bioinformatics*, 24(13):i375–i382, 2008.
- [42] J. Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*, volume 2 of *MPS/SIAM Series on Optimization*. SIAM, 2001.
- [43] R. T. Rockafellar. *Convex Analysis*, volume 28 of *Princeton Mathematics Series*. Princeton University Press, 1970.
- [44] C. Roos, T. Terlaky, and J.-Ph. Vial. *Interior Point Methods for Linear Optimization*. Springer Science, Heidelberg/Boston, 2006. (Note: This book is a significantly revised new edition of Interior Point Approach to Linear Optimization: Theory and Algorithms).
- [45] F. Sturm. Using SeDuMi 1.02: A Matlab toolbox for optimization over symmetric cones. *Optim. Methods Software*, 11-12:625–653, 1999.
- [46] W. Su, S. Boyd, and E. Candes. A differential equation for modeling nesterovs accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014.
- [47] A. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert, M.A. Gillette, A. Paulovich, S.L. Pomeroy, T.R. Golub, and E.S. Lander. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, 2005.
- [48] K.-Ch. Toh, M.J. Todd, and R.H. T’utüncü. On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. Tech. report, NUS Singapore, 2010.
- [49] Q. Tran Dinh, A. Kyrillidis, and V. Cevher. A proximal Newton framework for composite minimization: Graph learning without Cholesky decompositions and matrix inversions. In *Proceedings of The 30th International Conference on Machine Learning*, pages 271–279, 2013.
- [50] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. An inexact proximal path-following algorithm for constrained convex minimization. *SIAM J. Optim.*, 24(4):1718–1745, 2014.
- [51] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. Composite self-concordant minimization. *J. Mach. Learn. Res.*, 15:374–416, 2015.

- [52] R.H. Tütüncü, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.*, 95:189–217, 2003.
- [53] R. Vanderbei, H. Liu, L. Wang, and K. Lin. Optimization for compressed sensing: the simplex method and kronecker sparsification. *arXiv preprint arXiv:1312.4426*, 2013.
- [54] S.J. Wright. *Primal-Dual Interior-Point Methods*. SIAM Publications, Philadelphia, 1997.
- [55] H. Zhou, M.E. Sehl, J.S. Sinsheimer, and K. Lange. Association screening of common and rare genetic variants by penalized regression. *Bioinformatics*, 26(19):2375, 2010.